

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

До захисту допущено

Завідувач кафедри

(підпис) Віталій РОМАНКЕВИЧ
(ініціали, прізвище)

“ ____ ” _____ 202__ р.

Дипломний проєкт

на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Комп'ютерні системи та компоненти»
спеціальності 123 «Комп'ютерна інженерія»**

на тему: Програмні засоби каталогізації музичних творів

Виконав:

студент IV курсу, групи КВ-62
2(шифр групи)

Дученко Олександр Сергійович
(прізвище, ім'я, по батькові) _____ (підпис)

Керівник к. т. н., доцент Замятін Д.С.
(посада, науковий ступінь, вчене звання, прізвище та ініціали) _____ (підпис)

Консультант з нормоконтролю, доц.каф.СПСКС, к.т.н. Клятченко Я.М.
(назва розділу) (посада, вчене звання, науковий ступінь, прізвище, ініціали) _____ (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали) _____ (підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

ВІДОМІСТЬ ДИПЛОМНОГО ПРОЄКТУ

№ з/п	Формат	Позначення	Найменування	Кількість листів	Примітка
1	A4		Завдання на дипломний проект	2	
2	A4	ІАЛЦ. 045480.000 ОА	Опис альбому	2	
3	A1	ІАЛЦ. 045480.000 ТЗ	Технічне завдання	4	
4	A1	ІАЛЦ. 045480.000 ТП	Відомість технічного проекту	2	
5	A1	ІАЛЦ. 045480.000 ПЗ	Пояснювальна записка	55	
6	A1	ІАЛЦ. 045480.005 Д1	Читання з аудіо-файлу метаданих тегу ID3v2.3 Схема алгоритму	1	
7	A1	ІАЛЦ. 045480.006 Д2	Запис метаданих тегу ID3v2.3 до аудіо-файлу Схема алгоритму	1	
8	A1	ІАЛЦ. 045480.007 Д3	Загальна структура програми Схема структурна	1	
9	A1	ІАЛЦ. 045480.008 Д4	Загальний алгоритм програми Схема алгоритму	1	

				ІАЛЦ.045480.000		
	ПІБ	Підп.	Дата	Відомість дипломного проекту	Лист	Листів
Розробн.	Дученко О.С.				1	1
Керівн.	Замятін Д.С.				КПІ ім. Ігоря Сікорського Каф. СПіСКС Гр. КВ-62	
Консульт.						
Н/контр.						
Зав.каф.	Романкевич В.О.					

Пояснювальна записка до дипломного проєкту

на тему: Програмні засоби каталогізації музичних творів

Київ – 2020_ року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 123 «Комп'ютерна інженерія»

Освітньо-професійна програма «Комп'ютерні системи та компоненти»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Віталій РОМАНКЕВИЧ
(підпис) (ініціали, прізвище)

«__» _____ 20__ р.

ЗАВДАННЯ

на дипломний проєкт студента

Дученку Олександр Сергійовичу
(прізвище, ім'я, по батькові)

1. Тема проєкту «Програмні засоби каталогізації музичних творів», керівник проєкту Замятін Денис Станіславович, к. т. н., доцент, затверджені наказом по університету від «__» _____ 2020 р. № _____
2. Термін подання студентом проєкту 22 червня 2020 р.
3. Вихідні дані до проєкту див. Технічне завдання.
4. Зміст пояснювальної записки
 - аналіз існуючих програмних засобів для каталогізації музичних творів та редагування їх метаданих;
 - опис структури тегу метаданих ID3v2.3, та кадрів, які найчастіше використовуються;
 - опис алгоритмів читання та запису тегів ID3v2.3, реалізація розробленого програмного забезпечення;
 - тестування і аналіз результатів.

5. Перелік графічного матеріалу (із зазначенням обов'язкових креслеників, плакатів, презентацій тощо)

- читання з аудіо-файлу метаданих тегу ID3v2.3. Схема алгоритму;
- запис метаданих тегу ID3v2.3 до аудіо-файлу. Схема алгоритму;
- загальна структура програми. Схема структурна;
- загальний алгоритм програми. Схема алгоритму;
- слайди презентації.

6. Консультанти розділів проєкту*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 07.02.2020 р.

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	Вивчення літератури за тематикою проєкту	10.03.2020	
2.	Розроблення та узгодження технічного завдання	16.03.2020	
3.	Аналіз існуючих рішень	16.03.2020	
4.	Підготовка матеріалів першого розділу дипломного проєкту	17.04.2020	
5.	Підготовка матеріалів другого розділу дипломного проєкту	26.04.2020	
6.	Підготовка матеріалів третього розділу дипломного проєкту	10.05.2020	
7.	Підготовка матеріалів четвертого розділу дипломного проєкту	15.05.2020	
8.	Підготовка графічної частини дипломного проєкту	18.05.2020	
9.	Попередній захист дипломного проєкту	20.05.2020	

Студент

(підпис)

Олександр ДУЧЕНКО

Керівник проєкту

(підпис)

Денис ЗАМЯТІН

* Консультантом не може бути зазначено керівника дипломного проєкту.

АНОТАЦІЯ

Дипломний проєкт включає пояснювальну записку(50 с., 17 рис., 4 додатки).

Об'єкт розробки - програмний засіб для зручного та швидкого редагування метаданих музичних файлів, та їх подальшої каталогізації за певними критеріями.

Розроблений програмний засіб дозволяє:

- за допомогою вбудованого в програму файлового провідника завантажувати файли музичних творів для аналізу метаданих;
- редагувати метадані, вбудовані в завантажені файли;
- зберігати музичні файли з відредагованими метаданими за вибраним варіантом каталогізації.

Для виконання проєкту було проаналізовано структуру найпопулярнішого формату для зберігання метаданих, а саме, ID3 версії 2.3. Для аналізу та редагування кожного конкретного файлу з вбудованим кадром ID3v2.3 було написано програму. В процесі розробки було використано мову програмування C++ та середовище розробки MS Visual Studio. Для зручного використання програми було розроблено інтерфейс користувача з використанням API Direct2D.

Ключові слова: програмний засіб, каталогізація, музичний файл, редагування, метадані, кадр, ID3, інтерфейс користувача, C++, Direct2D.

ANNOTATION

The diploma project includes an explanatory note (50 p., 17 fig., 4 annexes).

The object of development is a software tool for easy and fast editing of metadata of music files, and their subsequent cataloging according to certain criteria.

The developed software allows you to:

- use the built-in file explorer to load music files for metadata analysis;
- edit metadata embedded in loaded files;
- save music files with edited metadata for the selected cataloging option.

To implement the project, the structure of the most popular metadata storage format was analyzed, namely, ID3 version 2.3. A program was written to analyze and edit each specific file with the ID3v2.3 built-in frame. The C ++ programming language and the MS Visual Studio development environment were used in the development process. A user interface has been developed using the Direct2D API for easy use of the program.

Keywords: software, cataloging, music file, editing, metadata, frame, ID3, user interface, C ++, Direct2D.

[illegible]

[illegible]

ЗМІСТ

1.	НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ.....	2
2.	ПІДСТАВА ДЛЯ РОЗРОБКИ.....	2
3.	МЕТА І ПРИЗНАЧЕННЯ РОБОТИ	2
4.	ДЖЕРЕЛА РОЗРОБКИ.....	2
5.	ТЕХНІЧНІ ВИМОГИ.....	2
5.1	Вимоги до програмного продукту, що розробляється.....	2
5.2	Вимоги до апаратного забезпечення	3
5.3	Вимоги до програмного та апаратного забезпечення користувача.....	3
6.	ЕТАПИ РОЗРОБКИ	4

					ІАЛЦ.045480.002 ТЗ				
Зм.	Арк.	№ докум.	Підп.	Дата					
Розроб.		Дученко О.С.			Програмні засоби каталогізації музичних творів Технічне завдання				
Перевір.		Замятін Д.С.							
Н. контр.		Клятченко Я.М.			Літ. Аркуш Аркушів КПІ ім. Ігоря Сікорського Каф. СПіСКС Гр. КВ-62				
Затв.		Романкевич В.О.							
							1	55	

1. НАЙМЕНУВАННЯ ТА ГАЛУЗЬ РОЗРОБКИ

Назва розробки: «Програмний засіб каталогізації музичних творів».

Галузь застосування: мультимедійне програмне забезпечення.

2. ПІДСТАВА ДЛЯ РОЗРОБКИ

Підставою для розробки є завдання на дипломне проектування на здобуття першого (бакалаврського) рівня вищої освіти, затверджене кафедрою системного програмування і спеціалізованих комп'ютерних систем Національного технічного університету України «Київський Політехнічний Інститут імені Ігоря Сікорського».

3. МЕТА І ПРИЗНАЧЕННЯ РОБОТИ

Метою даного проекту є створення програмного засобу для редагування метаданих формату ID3v2.3 у аудіо-файлах, та подальша їх каталогізація.

4. ДЖЕРЕЛА РОЗРОБКИ

Джерелом інформації є технічна та науково-технічна література, технічна документація, публікації у періодичних виданнях та електронні статті у мережі Інтернет.

5. ТЕХНІЧНІ ВИМОГИ

5.1 Вимоги до програмного продукту, що розробляється

- сумісність з операційною системою Windows;
- можливість завантаження аудіо-файлів за допомогою вбудованого файлового провідника;
- аналіз метаданих, вбудованих в завантажені файли;
- можливість редагування метаданих за допомогою зручного користувацького інтерфейсу;
- можливість зберігати аудіо-файли з відредагованими метаданими;

					ІАЛЦ. 045480.002 ТЗ	Арк.
						2
Змін.	Арк.	№ докум.	Підпис	Дата		

5.2 Вимоги до апаратного забезпечення

- Процесор: Intel Celeron G4900;
- Оперативна пам'ять: 4 Гб;

5.3 Вимоги до програмного та апаратного забезпечення користувача

- Операційна система Windows;

					ІАЛЦ. 045480.002 ТЗ	Арк.
						3
Змін.	Арк.	№ докум.	Підпис	Дата		

6. ЕТАПИ РОЗРОБКИ

№ з/п	Назва етапів виконання дипломного проекту	Термін виконання етапів
1.	Вивчення літератури за тематикою проекту	10.03.2020
2.	Розроблення та узгодження технічного завдання	16.03.2020
3.	Аналіз існуючих рішень	16.03.2020
4.	Підготовка матеріалів першого розділу дипломного проекту	17.04.2020
5.	Підготовка матеріалів другого розділу дипломного проекту	26.04.2020
6.	Підготовка матеріалів третього розділу дипломного проекту	10.05.2020
7.	Підготовка матеріалів четвертого розділу дипломного проекту	15.05.2020
8.	Підготовка графічної частини дипломного проекту	18.05.2020
9.	Попередній захист дипломного проекту	20.05.2020

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 045480.002 ТЗ

Арк.

4

[illegible]

[illegible]

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	5
1. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ КАТАЛОГІЗАЦІЇ МУЗИЧНИХ ТВОРІВ ТА РЕДАГУВАННЯ ЇХ МЕТАДАНИХ	6
1.1. Аналіз форматів музичних файлів та метаданих	6
1.2 Аналіз існуючих рішень	10
1.3 Обґрунтування теми дипломного проекту.	15
1.4 Обґрунтування вибору середовища розробки і мови програмування.....	16
2. ОПИС СТРУКТУРИ ТЕГУ МЕТАДАНИХ ID3v2.3, ТА КАДРІВ, ЯКІ НАЙЧАСТІШЕ ВИКОРИСТОВУЮТЬСЯ.....	18
2.1 Основна структура тегу метаданих формату ID3v2.3	18
2.2 Структура кадру ID3v2.3	20
2.3 Перелік типів кадрів, які найчастіше використовуються, їх ідентифікаторів, полів які вони зберігають, та їх призначення.....	23
2.3.1 Кадр унікального ідентифікатору файлу UFID	23
2.3.2 Текстові кадри T000-TZZZ, за винятком TXXX	24
2.3.3 Текстовий кадр TXXX	29
2.3.4 Кадри URL-посилань W000-WZZZ, за винятком WXXX.....	29
2.3.5 Кадр URL-посилань WXXX	31
2.3.6 Кадр списку залучених людей IPLS	32
2.3.7 Кадр списку кодів часових подій ETCO	32
2.3.8 Кадр несинхронізованого тексту пісні USLT	34
2.3.9 Кадр синхронізованого тексту пісні SYLT	35

					ІАЛЦ.045480.002 ТЗ			
Зм.	Арк.	№ докум.	Підп.	Дата				
Розроб.		Дученко О.С.			Програмні засоби каталогізації музичних творів Технічне завдання	Літ.	Аркуш	Аркушів
Перевір.		Замятін Д.С.					1	55
						КПІ ім. Ігоря Сікорського Каф. СПіСКС Гр. KB-62		
Н. контр.		Клятченко Я.М.						
Затв.		Романкевич В.О.						

2.3.10	Кадр коментарів COMM	36
2.3.11	Кадр вкладеного зображення APIC	37
3.	ОПИС АЛГОРИТМІВ ЧИТАННЯ ТА ЗАПISУ ТЕГІВ ID3v2.3, РЕАЛІЗАЦІЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	41
3.1	Алгоритм читання тегів	41
3.2	Алгоритм запису тегів	42
3.3	Загальний опис розробленого програмного забезпечення	43
3.4	Структури даних, використані в розробленому програмному забезпеченні.....	47
4.	ТЕСТУВАННЯ І АНАЛІЗ РЕЗУЛЬТАТІВ	50
	ВИСНОВКИ.....	54
	СПИСОК ВИКОРИСТНОЇ ЛІТЕРАТУРИ	55

ДОДАТКИ

Додаток 1. Копії графічних матеріалів

- ІАЛЦ.045480.005 Д1. Читання з аудіо-файлу метаданих тегу ID3v2.3. Схема алгоритму.
- ІАЛЦ.045480.006 Д2. Запис метаданих тегу ID3v2.3 до аудіо-файлу. Схема алгоритму.
- ІАЛЦ.045480.007 Д3. Загальна структура програми. Схема структурна.
- ІАЛЦ.045480.008 Д4. Загальний алгоритм програми. Схема алгоритму.

Додаток 2. Презентація

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

C++ – високорівнева мова програмування;

MP3 – найрозповсюдженіший формат файлів для зберігання звуку;

Метадані – це дані, що пояснюють, або описують інші дані;

ID3 (Identify a MP3) – формат метаданих, який найчастіше використовується разом з форматом для зберігання аудіоінформації MP3;

ID3v2 – другий стандарт формату ID3;

ID3v2.3 – третя версія стандарту ID3v2;

API (Application Programming Interface) – інтерфейс програмування додатків;

Direct2D – 2D інтерфейс програмування векторної графічної програми. Дозволяє відображати зображення з файлів, та примітиви, такі як, лінії, прямокутники, еліпси тощо;

DirectWrite – API для форматування і відображення тексту на екрані;

ASCII (American Standard Code for Information Interchange) – система кодування символів;

Unicode – стандарт кодування, розроблений для забезпечення можливості представити спеціальні символи та всі писемності світу у вигляді, з яким можуть працювати комп'ютери;

Filesystem – бібліотека додана в стандарті C++17, яка дозволяє виконувати певні операції над файловою системою комп'ютера та її компонентами, такими як, шляхи, файли, папки.

STL (Standard Template Library) – стандартна бібліотека шаблонів в мові програмування C++, яка містить набір контейнерів, засобів доступу до їхнього вмісту і різних допоміжних функцій;

Vector – послідовний контейнер що зберігає масив змінної довжини. Компонент STL;

Map – асоціативний контейнер, що зберігає набір даних формату ключ – значення. Компонент STL;

Wstring – клас для зберігання строкової інформації, представленої символами розміром 2 байти;

string – клас для зберігання строкової інформації, представленої символами розміром 1 байт;

Дискретизація – перетворення неперервних функцій в сукупність її значень при різних наборах аргументів;

Частота дискретизації – при перетворенні неперервного сигналу в дискретний визначає кількість сигналів за секунду;

АЦП – аналого-цифровий перетворювач;

ЦАП – цифро-аналоговий перетворювач;

ПЗ – програмне забезпечення;

%XXXXXXXX – позначення байту у вигляді бітів, де X може бути лише 1 або 0;

%X – позначення біту з невідомим значенням;

\$XX – позначення байту у шістнадцятковому вигляді, де X може бути цифрою, або літерою від A до F;

\$X – позначення байту з невідомим значенням;

«» – за допомогою лапок в Розділі 2 можуть позначатися строки, які записано, або зчитано з файлу;

CRC-32 (Cyclic redundancy check) – алгоритм, для виявлення помилок в збережених даних. Обчислює контрольну суму;

Тег – інформаційна мітка, що міститься в аудіо-файлі для його опису.

ВСТУП

В епоху обчислювальної техніки музика стала дуже розповсюдженою. Раніше люди не могли слухати музику в дорозі, оскільки вони зберігалась або відтворювалась на пристроях, які неможливо було взяти з собою. Але, з плином часу, технології розвивались, і засоби збереження і відтворення звукової інформації ставали все мініатюрніше. В наші дні вони з легкістю поміщаються у кишені, і велика кількість людей слухає музику під час ранкової пробіжки, по дорозі з або на роботу, вдома під час відпочинку, або повсякденних справ. У багатьох з нас музичні файли зберігаються на цифрових носіях інформації, таких як, USB носії, SD, або microSD карти, жорсткі диски комп'ютерів тощо. Найрозповсюдженішим типом музичних файлів є MP3. Для зберігання опису файлу до нього дописують окрему частину – поле, що містить метадані. Звук записаний у файли відтворюється спеціальною програмою – музичним програвачем. З плином часу може виникнути потреба змінити музичний програвач. Тут і виникає проблема: якщо початковий файл не мав вбудованих метаданих, і користувач редагував назви пісень, альбомів, груп, імена виконавців тощо, можлива така ситуація, що ці зміни не були занесені до початкового файлу і будуть відображатися лише у використовуваній раніше програмі. В такому випадку, в результаті зміни програвача на інший, дані про музичні твори буде втрачено. Рішенням цієї проблеми повинне стати програмне забезпечення, розроблене в ході виконання даного дипломного проєкту.

Завданням дипломного проєкту є розробка програмного забезпечення для каталогізації музичних творів за вибраним параметрами, та редагування і збереження метаданих, конкретних файлів. Каталогізація, наприклад, може бути виконана у вигляді папок з назвами виконавців, або музичних груп, які, в свою чергу будуть містити безпосередньо файли музичних творів. Таким чином, користувач матиме можливість один раз задати всі потрібні йому дані безпосередньо в файли музичних творів і не перейматися за можливу їх втрату в майбутньому.

1. АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ КАТАЛОГІЗАЦІЇ МУЗИЧНИХ ТВОРІВ ТА РЕДАГУВАННЯ ЇХ МЕТАДАНИХ

1.1. Аналіз форматів музичних файлів та метаданих

Дуже давно люди почали зберігати звуки для подальшого відтворення. Спершу звук записувався на вінілових платівках. На заміну платівкам прийшли магнітні стрічки. Далі збереження звуку відбувалось з використанням компакт дисків. Але для запису на компакт диски звук довелося відцифровувати. Для відцифровування було створено декілька типів так званих аналого-цифрових перетворювачів. Це пристрої, задачею яких є перетворення вхідного аналогового сигналу в цифровий сигнал, який певним чином характеризує амплітуду аналогового сигналу. На сьогоднішній день більшість з перелічених носіїв вже не використовуються, натомість звук відцифровують та зберігають у вигляді файлів на жорстких дисках комп'ютерів, внутрішній пам'яті смартфона, SD або microSD картах, USB носіях тощо. Фізичні копії було досить легко каталогізувати, оскільки на вінілову платівку, касету, або компакт диск наклеєно плівку з усією необхідною інформацією. Каталогізація файлів можлива на рівні файлової системи, та за допомогою даних, записаних в кожен музичний файл додатковим полем.

На даний момент світі існує дуже багато форматів файлів, що використовуються для зберігання відцифрованого звуку.

Аудіо-файли поділяються на три основні категорії:

- без стиснення;
- зі стисненням, але без втрат якості;
- зі стисненням і втратами якості.

Прикладами форматів без стиснення є WAV, AIFF, DSD.

WAV – досить старий формат, розроблений компаніями Microsoft та IBM в далекому 1991 році. Насправді, цей формат можна віднести і до категорії зі стисненими даними, оскільки це контейнер. Але найчастіше в ньому зберігається не стиснений звук в форматі PCM, тому будемо вважати, що цей формат не

					ІАЛЦ. 045480.004 ПЗ	Арк.
						6
Змін.	Арк.	№ докум.	Підпис	Дата		

стискає звукові дані. PCM – імпульсно кодова модуляція. Це процес перетворення аналогового сигналу у цифровий. Під час перетворення через певні інтервали часу вимірюється амплітуда аналогового сигналу, яка кодується цифрами. На одну хвилину запису використовується приблизно 32 мегабайти пам'яті.

AIFF – формат, розроблений Apple 1988 році на основі формату IFF. Хвилина запису займає приблизно 10 мегабайтів.

DSD – формат, який використовує набагато більшу частоту дискретизації, ніж всі інші формати, а саме - від 64*44.1кГц до 256*44.1 кГц, при цьому використовуючи роздільну здатність рівну лише одному біту, тоді як інші формати використовують роздільну здатність приблизно 16-32, в залежності від формату. Наведені формати здебільшого використовуються для професійного звукозапису.

Прикладами форматів зі стисненням без втрат якості є FLAC, ALAC, WMA Lossless.

FLAC – розроблений у 2001 році. Досить популярний. Підтримується великою кількістю програм.

ALAC – розроблений приблизно у 2004 році. Є відкритим аудіо-кодеком, дані якого зберігають в контейнері MP4 з розширенням .m4a.

WMA Lossless – представлений в 2003 році. Остання версія датується 2008 роком. Використовує розширення .wma.

Прикладами форматів зі стисканням та втратами якості є MP3, Ogg Vorbis, AAC тощо.

MP3 – один із найпопулярніших форматів зберігання звуку на сьогоднішній день. Вперше з'явився ще в 1993 році. Розроблений таким чином, щоб не зберігати ту частину звукової інформації, яка не сприймається вухом людини. Хвилина стисненого звуку буде займати приблизно 2 мегабайти.

Ogg Vorbis – вважається другим за популярністю після MP3. Розроблений у 1998 році і випущений у 2002 році. Підтримується великою кількістю програм.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 045480.004 ПЗ

Арк.

7

АСС – розроблена у 1997 році, як аналог формату MP3. Трохи краще передає звук збільшним ступенем стиснення, ніж MP3. Як і попередні формати, підтримується великою кількістю програм.

Було наведено найрозповсюджені формати, що зберігають відцифрований звук. Кожен з цих форматів має право на життя і використовується в конкретних сферах. Це зумовлено тим, що записаний звук використовується у великій кількості напрямків, починаючи від кіноіндустрії, і закінчуючи радіомовленням. Для отримання звуку високої якості необхідне дуже дороге обладнання, яке не простіше взяти в оренду, ніж купити. Саме тому для запису зазвичай використовують так звані студії звукозапису. Це спеціально створене, і обладнане приміщення, в якому записують та обробляють звук у високій якості.

Але не тільки для запису звуку у високій якості потрібне коштовне обладнання. Для відтворення звуку у високій якості необхідно мати цифро-аналоговий перетворювач та акустичні систему високої якості. У середньостатистичного користувача нема цих технічних засобів. Це зумовлено тим, що в сучасні смартфони, планшети, кишенькові програвачі, материнські плати комп'ютерів та ноутбуків, або недорогі підсилювачі, не спрямовані на відтворення звуку високої якості, на відміну від дорогих систем, які не використовуються середньостатистичним користувачем. Також, для зберігання звуку у високій якості потрібний великий обсяг вільної пам'яті на пристрої збереження інформації.

Саме через ці причини, дуже великої популярності серед середньостатистичних користувачів досягли формати збереження аудіоінформації, що використовують стиснення звуку з втратами його якості. Користувач, що прослуховує музику через смартфон або комп'ютер, використовуючи недорогі гарнітуру, або акустичну систему, не почує різниці між звуком записаним у форматі високої якості, та стисненим з втратами якості. Але при цьому на носій інформації деякого конкретного розміру можна буде записати більшу кількість звукових файлів, якщо використовувати формати зі стисненням та втратою якості звуку.

					ІАЛЦ. 045480.004 ПЗ	Арк.
						8
Змін.	Арк.	№ докум.	Підпис	Дата		

Історично склалося, що найпопулярнішим серед середньостатистичних користувачів став формат MP3, оскільки він стискає звукові дані досить ефективно, при цьому дуже мало знижуючи якість збереженого звуку.

Для ідентифікації музичних творів збережених у вигляді файлів можливо використовувати два підходи:

- записати необхідну інформацію в ім'я файлу;
- дописати безпосередньо в файл спеціальну частину, яка буде містити необхідну інформацію.

Перший варіант може дати дуже обмежену функціональність. Наприклад, в назву файлу неможливо буде додати картинку, записати текст пісні, або записати якісь коментарі. Тому цей варіант можна використати тільки для загального опису деякого файлу.

Другий варіант дає значно більші можливості для опису конкретного файлу. Дані, що записуються в конкретний файл, як доповнення до основних даних файлу, називаються метаданими. Вони бувають різних типів, а також по різному підтримуються форматами музичних файлів. Деякі формати метаданих, які існують на сьогоднішній день є:

- ID3v1;
- ID3v2;
- коментарі Vorbis;
- APEv2.

Формат ID3v1 – розроблений у 1996 році для формату MP3. Перша версія була розміром лише 128 байтів, що починались літерами TAG. Для підтримки старими програвачами було вирішено помістити метадані в кінець файлу. Основним недоліком був обмежений розмір поля даних, якого не вистачало для зберігання більшості інформації. Жанр було закріплено за числом. Також значним недоліком було збереження строк згідно стандарту ISO 8859-1. Це не давало змогу записувати нестандартні символи, наприклад, літери деяких мов. У 1997 році формат було розширено, але цього все одно не вистачало.

					ІАЛЦ. 045480.004 ПЗ	Арк.
						9
Змін.	Арк.	№ докум.	Підпис	Дата		

На зміну цьому формату прийшов, абсолютно інший за структурою, формат ID3v2.

Формат ID3v2 – опублікований у 1998 році. На відміну від старого формату ID3v1, має змінну довжину і може розташовуватися як на початку, так і в кінці файлу. Складається з окремих частин, які несуть в собі службову і основну інформацію. Ці частини називають фреймами. Формат має декілька типів фреймів, які можуть містити у собі різну інформацію, таку як, текст, посилання, зображення тощо. На сьогоднішній день є найрозповсюдженішим форматом метаданих, які використовуються в музичних файлах формату MP3.

APeV2 – формат метаданих, який є удосконаленою версією формату APeV1, розробленого спеціально для формату Monkey's Audio. Даний формат підтримується у якості поля метаданих лише деякими музичними програвачами. На даний момент майже не використовується.

Коментарі Vorbis підтримуються у форматах OGG Vorbis, FLAC та деяких інших. Призначений для зберігання коротких текстових коментарів. В даному форматі не можна зберігати двійкові дані. На відміну від формату ID3 не містить наперед заданих форматів для значення змінних, та дозволяє використовувати будь які імена тегів.

Слід зазначити, що може бути важко або неможливо прочитати метадані, збережені в файлі без музичного програвача, або спеціальних програм.

1.2 Аналіз існуючих рішень

З самого початку історії програмування, воно полегшувало виконання певних задач для людей. Спочатку, програмування лише допомагало завантажити в обчислювальну систему дані та правила для рішення поставленої математичної задачі. Пізніше було створено персональні комп'ютери, і для полегшення взаємодії користувача з комп'ютером було розроблено операційну систему. Користувацький інтерфейс постійно покращується, і зараз майже

					ІАЛЦ. 045480.004 ПЗ	Арк.
						10
Змін.	Арк.	№ докум.	Підпис	Дата		

неможливо уявити програму без інтерфейсу, яка використовує лише командний рядок. Програма повинна мати простий у використанні інтерфейс.

Але, з розвитком програмування розвивалось і шахрайство. На сьогоднішній день існує величезна кількість вірусів і шкідливого програмного забезпечення. Кожен завантажений з інтернету файл, і тим паче програма можуть містити в собі додаток у вигляді шкідливого ПЗ.

Першим проаналізованим аналогом є веб-ресурс для редагування метаданих під назвою tagmp3.net. Він дуже простий у використанні. Необхідно лише вибрати потрібний файл з файлового провідника, та дочекатись його завантаження на сервер ресурсу. Це можна побачить на рис.1.1. Далі автоматично завантажується сторінка для редагування метаданих, яку можна побачить на рис.1.2.

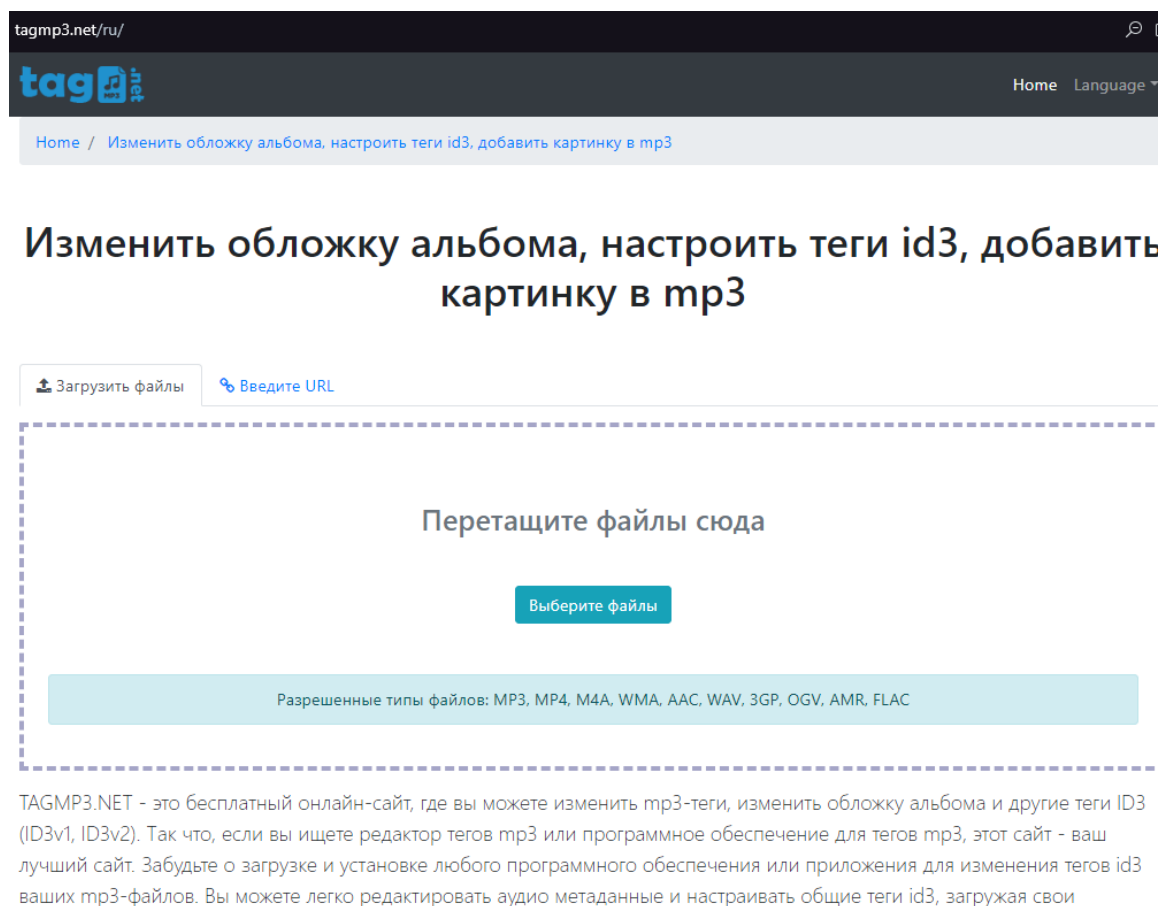



Рисунок 1.1 – Завантаження файлу для подальшого редагування метаданих.

1 04.назва файлу.mp3

Существующее оформление альбома



Выберите обложку нового альбома / картинку

Файл не выбран

Browse

заглавие

Назва

художник

Виконавець

Альбом

Fallen

Номер дорожки

04/12

Жанр

жанр

Комментарии

IMA-Sound

Год

2003

Рисунок 1.2 – Сторінка редагування метаданих завантаженого файлу ресурсу tagmp3.net.

До недоліків цього ресурсу можна віднести:

- необхідність завантаження файлу на віддалений сервер;
- обмежений перелік полів метаданих для редагування.

Переваги:

- зручний інтерфейс;
- підтримка великої кількості форматів музичних файлів;
- локалізація ресурсу для різних регіонів.

Необхідність завантаження файлу на віддалений сервер є суттєвим недоліком даного ресурсу, оскільки при великих кількостях файлів, які потребують редагування, значний час піде на надсилання та завантаження файлів.

Даний ресурс буде гарною альтернативою, якщо необхідно відредагувати лише кілька музичних файлів.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 045480.004 ПЗ

Арк.

12

Другим проаналізованим аналогом є програма Mp3tag. Досить проста у використанні, має зрозумілий інтерфейс рис.1.3.

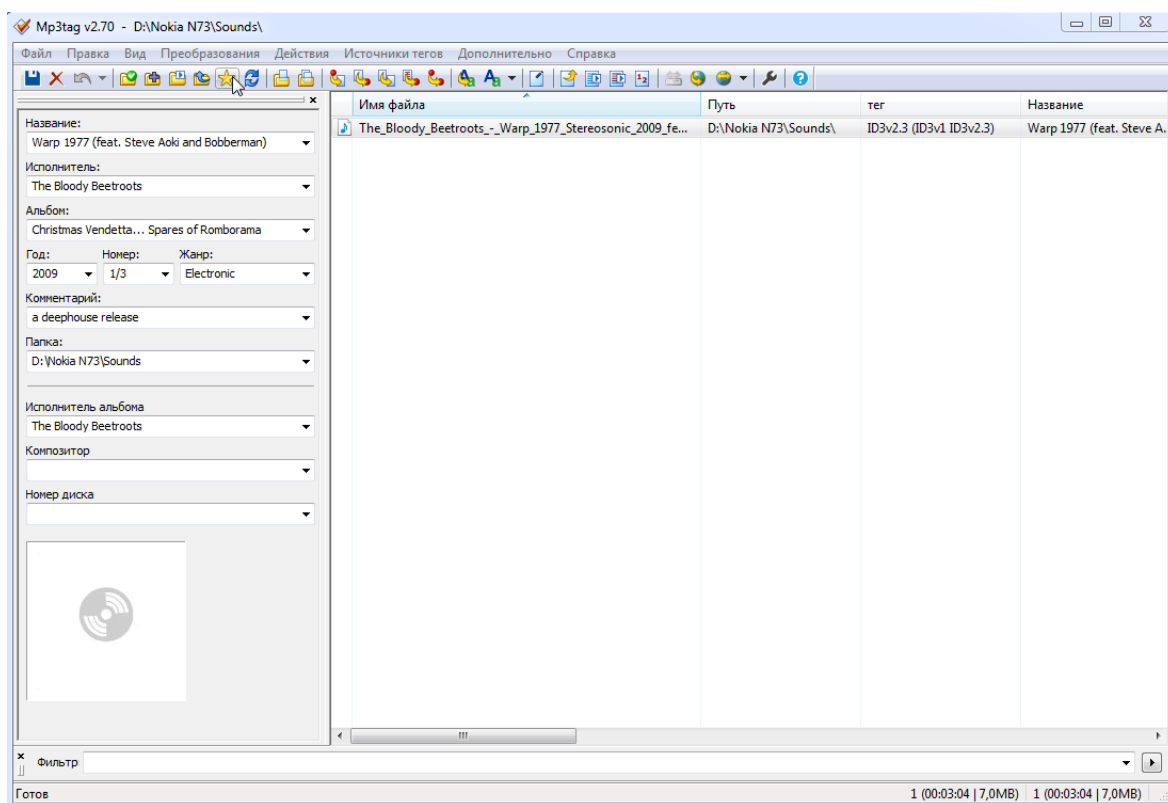


Рисунок 1.3 – Інтерфейс програми Mp3tag.

До недоліків даного програмного забезпечення можна віднести:

- неможливість відтворювати музичні файли;
- обмежений перелік полів метаданих для редагування.
- можлива наявність вбудованого шкідливого програмного забезпечення.

Переваги:

- зручний інтерфейс;
- локалізація програмного забезпечення для різних регіонів;
- підтримка великої кількості форматів музичних файлів.

Необхідність завантаження і встановлення даного програмного забезпечення несе потенційну небезпеку зараженням комп'ютера користувача шкідливим програмним забезпеченням.

Дане програмне забезпечення може бути непоганою альтернативою, якщо необхідно відредагувати досить велику кількість музичних файлів.

					ІАЛЦ. 045480.004 ПЗ	Арк.
Змін.	Арк.	№ докум.	Підпис	Дата		13

Третім проаналізованим аналогом є програма Tag&Rename. Має не дуже зрозумілий інтерфейс рис.1.4.

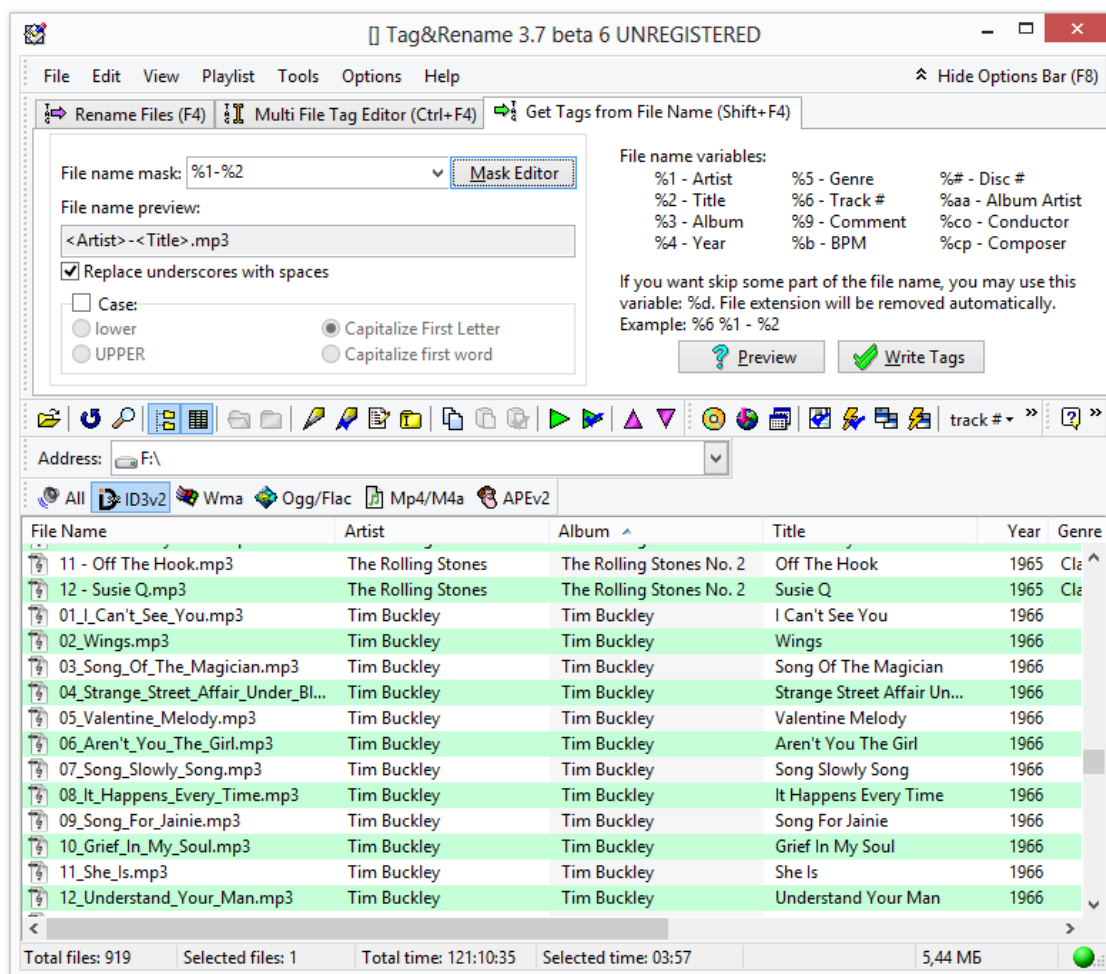


Рисунок 1.4 – інтерфейс програми Tag&Rename.

До недоліків даного програмного забезпечення можна віднести:

- неможливість відтворювати музичні файли;
- не дуже зручний інтерфейс;
- можлива наявність вбудованого шкідливого програмного забезпечення;
- необхідність оплати продукту після закінчення пробного терміну.

Переваги:

- можливість згенерувати деякі поля метаданих з імені файлу;
- підтримка великої кількості форматів музичних файлів;
- можливість редагування великої кількості полів метаданих.

На початку роботи інтерфейс буде не дуже зрозумілим. Як і в попередньому варіанті, може мати вбудоване шкідливе програмне забезпечення.

Якщо ж припустити, що дане програмне забезпечення не містить вірусів, його можна вважати найкращим з представленою вище. Воно надає досить непогану функціональну частину для каталогізації і редагування метаданих.

1.3 Обґрунтування теми дипломного проєкту

В результаті технічного прогресу, і роботи великої кількості людей, було розроблено багато форматів для зберігання та опису звукової інформації. Всі вони розроблені з певною метою. Деякі зберігають звукову інформацію без зниження якості, а деякі, непомітно для великої кількості людей, дуже сильно зменшують розмір інформації, яку необхідно записати на пристрій. Найпопулярнішим форматом для зберігання звуку став формат MP3. Найпопулярнішим форматом для опису даних, збережених у форматі MP3 став формат метаданих ID3v2.

Для редагування метаданих та каталогізації музичних творів різними авторами та компаніями було розроблено власні програми. У всіх програм є певні переваги і недоліки. Частина програм є безкоштовними. Але ми не можемо бути впевненими у чесності розробників, оскільки вони могли вбудувати в свої програми шкідливе програмне забезпечення.

Тому темою дипломного проєкту було обрано створення програмного засобу для редагування метаданих та подальшої каталогізації з такими перевагами:

- простота у використанні і зручний інтерфейс;
- безкоштовне програмне забезпечення з відкритим кодом;
- робота з великою кількістю файлів;
- можливість відтворювати музичні файли.

Даний програмний засіб можна буде використовувати вже в готовому вигляді, або модифікувати її на свій розсуд.

					ІАЛЦ. 045480.004 ПЗ	Арк.
						15
Змін.	Арк.	№ докум.	Підпис	Дата		

1.4 Обґрунтування вибору середовища розробки і мови програмування

Для створення програмного забезпечення для аналізу, редагування і збереження метаданих аудіо-файлів формату MP3, та подальшої їх каталогізації було вибрано високорівневу мову програмування C++ з використанням графічного API Direct2D для розробки користувацького інтерфейсу. C++ на даний момент є досить популярною мовою програмування для розробки прикладних програм. Дана мова є компільованою, і використовує строгу типізацію даних. Підтримує три парадигми програмування: процедурну, узагальнену та об'єктно-орієнтовану. Була розроблена у 1983, але продовжує розвиватися, приблизно раз у три роки оновлюючи стандарт. Поточний стандарт – C++17. Однією з особливостей цього стандарту є запровадження бібліотеки для роботи з файловою системою. Ця бібліотека називається `filesystem` і була використана в ході написання програмного забезпечення згідно з темою дипломного проєкту. Завдяки ній в програмі реалізовано можливість доступу та навігації по файловій системі комп'ютера. Варто зазначити, що C++ має велику кількість документації з поясненнями, і прикладами використання. Це значно полегшує розробку програмних засобів даною мовою. В даній мові програмування підтримуються зручні читання та запис файлів, що полегшує роботу з метаданими, вбудованими в музичні файли.

Бібліотека Direct2D також була вибрана, оскільки вона має велику кількість офіційної документації з прикладами коду. Також великий вплив на вибір саме цієї бібліотеки були такі переваги:

- швидкодія;
- можливість зручно працювати з примітивами;
- можливість зручно працювати з відображенням зображень, збережених у файлах.

У якості середовища для розробки програмного забезпечення було вибрано MS Visual Studio. Це одна з найсучасніших і найрозповсюдженіших середовищ розробки для мови C++. Вона постійно оновлюється, і є дуже зручною для

використання. Перевагами даного середовища розробки є:

- швидка робота компілятора;
- зручний і зрозумілий інтерфейс;
- зручний відлагоджувач програм;
- можливість перегляду стеку викликів під час відлагоджування;
- можливість детального перегляду ресурсів, використаних програмою, що відлагоджується на даний момент.

Висновки

В даному розділі було розібрано найпопулярніші формати для збереження звукових даних, та їх переваги і недоліки. Найпопулярнішим виявився формат MP3. Також було розглянуто формати метаданих, серед яких по розповсюдженню перше місце займає ID3v2. Програмне забезпечення, написане в ході дипломного проєкту повинно мати можливість роботи саме з цими форматами, для охоплення більшої аудиторії. Далі було проаналізовано аналоги розроблюваного програмного забезпечення. Кожен з них має свої переваги і недоліки. Було прийняте рішення розробити програмне забезпечення з переліченими у розділі 1.3 перевагами. Останнім був вибір мови програмування та середовища розробки. Зважаючи на досить велику популярність, простоту, функціональність і постійний розвиток, було вибрано мову C++. Одним з найкращих середовищ для розробки мовою C++ на сьогоднішній день є MS Visual Studio.

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 045480.004 ПЗ

Арк.

17

2. ОПИС СТРУКТУРИ ТЕГУ МЕТАДАНИХ ID3v2.3, ТА КАДРІВ, ЯКІ НАЙЧАСТІШЕ ВИКОРИСТОВУЮТЬСЯ

2.1 Основна структура тегу метаданих формату ID3v2.3

Формат метаданих ID3v2.3 є найрозповсюдженішою версією формату ID3v2.

Заголовок тегу ID3v2 повинен бути першою інформацією, записаною у файлі і має розмір 10 байт, які складаються з:

1. ідентифікатору ID3v2, який складається з 3 байтів «ID3»;
2. версії ID3v2, який складається з 2 байтів \$ 03 00;
3. прапори, який складається з 1 байту, в якому використано старші три біти %xyz00000;
4. розмір тегу ID3v2, який складається з 4 байтів, в кожному з яких використано всі, окрім старшого, біти 4 * 0xxxxxxx.

Перші три байти тегу завжди встановлені «ID3» для ідентифікації тегу.

Далі йдуть байти версії. Перший байт версії є основною версією, а другий – ревізією. Всі ревізії зворотно сумісні, тоді як, основні версії – ні. Версія та ревізія ніколи не будуть мати значення \$FF.

За байтами версії йде один байт для поля прапорців, з яких використовуються тільки три старші біти, які позначають:

1. сьомий біт %x позначає, чи використовується розсинхронізація;
2. шостий біт %y позначає, чи йде за основним заголовком розширений;
3. п'ятий біт %z використовується як показник експериментального заголовку.

Всі інші біти поля прапорців мають бути встановлені в 0. Якщо один з цих бітів буде встановлений, тег може бути проігноровано аналізатором.

Далі слідує 4 байти, які містять загальний розмір тегу без 10 байтів заголовку. В цей розмір включаються безпосередньо розмір даних та розмір так званого «відступу». Відступ – спеціальна частина тегу, яка заповнена 0. Призначенням відступу є можливість додаткового запису деякої кількості

					ІАЛЦ. 045480.004 ПЗ	Арк.
						18
Змін.	Арк.	№ докум.	Підпис	Дата		

інформації в тег без перезапису всього файлу. Кодування розміру відбувається в сім з восьми бітів кожного байту. Старший біт кожного байту завжди має бути встановленим в 0. Тобто, сумарна кількість біт, які можна використовувати для збереження розміру тегу – 28. Вони дають змогу зберегти до 256 мегабайтів інформації.

00000255	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	49	44	33	03	00	00	00	00	1f	76	54	41	4c	42	00	00	ID3.....vTALB..
00000010	00	0f	00	00	01	ff	fe	10	04	3b	04	4c	04	31	04	3eяю...;.L.l.>
00000020	04	3c	04	54	43	4f	4e	00	00	00	0d	00	00	01	ff	fe	.<.TCON.....яю
00000030	21	04	42	04	38	04	3b	04	4c	04	54	49	54	32	00	00	!.B.8.;.L.TIT2..
00000040	00	0d	00	00	01	ff	fe	1d	04	30	04	37	04	32	04	30яю..0.7.2.0
00000050	04	54	50	45	31	00	00	00	17	00	00	01	ff	fe	12	04	.TPE1.....яю..
00000060	38	04	3a	04	3e	04	3d	04	30	04	32	04	35	04	46	04	8...>.=.0.2.5.F.
00000070	4c	04	54	50	4f	53	00	00	00	05	00	00	01	ff	fe	32	L.TPOS.....яю2
00000080	00	54	52	43	4b	00	00	00	0b	00	00	01	ff	fe	31	00	.TRCK.....яю1.
00000090	2f	00	31	00	34	00	54	59	45	52	00	00	00	0b	00	00	/.1.4.TYER.....
000000a0	01	ff	fe	32	00	30	00	32	00	30	00	55	53	4c	54	00	..яю2.0.2.0.USLT.
000000b0	00	00	20	00	00	01	65	6e	67	ff	fe	00	00	ff	fe	22engяю..яю"
000000c0	04	35	04	3a	04	41	04	42	04	20	00	3f	04	56	04	41	.5...A.B. .?.V.A
000000d0	04	3d	04	56	04	00	00	00	00	00	00	00	00	00	00	00	.=.V.....
000000e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Рисунок 2.1 – Загальна структура тегу версії ID3v2.3.

Розглянемо загальну структуру тегу на прикладі реального файлу рис.2.1:

1. Червоним кольором виділено байти для ідентифікації тегу метаданих формату ID3v2.
2. Помаранчевим кольором виділено байти для ідентифікації версії та ревізії формату ID3v2. В даному прикладі версія має значення 3, ревізія – 0.
3. Зеленим кольором виділено байт прапорців. В даному прикладі він має значення за замовчуванням – 0, тобто всі прапорці неактивні.
4. Синім кольором виділено байти розміру тегу. В даному прикладі два старші байти записано у якості нулів, а два молодші – 0001111 та 01110110 . Значення кожен байту необхідно помножити на 128 в степені від 3 до 0, в залежності від порядкового номеру байту справа наліво. Тому два старші байти будуть дорівнювати $0 \cdot 128^3$ та $0 \cdot 128^2$, а два молодші – $31(1F_{16}) \cdot 128^1$ та $118(76_{16}) \cdot 128^0$. Для отримання розміру необхідно скласти всі ці значення. Для даного прикладу розмір буде дорівнювати $3968 + 118 = 4086$ байтів.

5. Темно-сірим кольором виділено байти кадрів, як містять в собі метадані.

6. Фіолетовим кольором виділено байти відступу.

Варто зауважити, що тег може містити додатковий розширений заголовок. Інформація, яка в нього записана є неважливою для правильного розбору інформації тегу, отже розширений заголовок не є обов'язковим. Здебільшого його не використовують. В прикладі не демонструється розширений заголовок. Він складається з таких полів:

1. розмір розширеного заголовку, записаний в 4 байтах;
2. прапорці розширеного заголовку, записані в 2 байтах;
3. розмір «відступу».

В полі прапорців розширеного заголовку для формату ID3v2.3 використовується лише старший біт старшого байту, тоді як всі інші мають бути встановленими в 0. Якщо цей прапорець встановлений, до розширеного заголовку додаються 4 байти даних CRC-32.

2.2 Структура кадру ID3v2.3

В форматі ID3v2 метадані мають досить жорстку структуру. Тег містить спеціальні кадри, які містять певний набір службової та корисної інформації. Кадри бувають з різними ідентифікаторами, там мають змінну довжину. Кожен кадр складається з заголовку і полів даних. Заголовок кожного кадру є службовою інформацією кадру, і складається з таких полів даних:

1. поле ідентифікатор кадру, складається з 4 байтів;
2. поле розміру кадру, складається з 4 байтів;
3. поле прапорців, складається з 2 байтів.

Від ідентифікатору кадру залежить те, які поля він використовує, можливість кодування даних в цих полях, можливість повторного використання того чи іншого типу. Ідентифікатор кадру складається з чотирьох символів, якими можуть бути великі літери від A до Z та цифри від 0 до 9. Ідентифікатори,

					ІАЛЦ. 045480.004 ПЗ	Арк.
						20
Змін.	Арк.	№ докум.	Підпис	Дата		

що починаються з “X”, “Y”, або “Z”, призначені для експериментального використання, і кожен може використовувати їх як забажає. Слід пам’ятати, що хтось міг вже використати обраний вами ідентифікатор. Всі інші ідентифікатори вже використовуються, або зарезервовані для подальшого використання.

Поле прапорців складається з трьох байтів, в кожному з яких під прапори відведено по 3 старші біти кожного з байтів. Структура прапорців виглядає наступним чином: %abc00000 %xyz00000.

Прапорець “a” відповідає за надання інформації програмному забезпеченню про те, що робити з кадром у випадку, якщо він невідомий і тег будь-яким чином змінився. Це стосується всіх видів змін, включаючи розширення відступу, та упорядкування кадрів. Значення прапорця відповідають:

- 0 –зберегти кадр;
- 1 –відкинути кадр.

Прапорець “b” відповідає за надання інформації програмному забезпеченню про те, що робити з кадром у випадку, якщо він невідомий і файл, виключаючи тег змінився. Це не застосовується, коли аудіо повністю замінено іншими аудіо-даними. Значення прапорця відповідають:

- 0 – зберегти кадр;
- 1 – відкинути кадр.

Прапорець “c” відповідає за надання інформації програмному забезпеченню про те, що вміст цього кадру призначений лише для читання, і зміна вмісту може щось порушити. Якщо зміст було змінено, не знаючи, чому фрейм позначений для читання, і не використовуючи належних засобів для компенсації, прапорець слід очистити.

Прапорець “x” відповідає за надання інформації програмному забезпеченню про те, чи стиснено кадр.

- 0 – кадр не стиснено;
- 1 – кадр стиснено за допомогою ZLIB з 4 байтами для “декомпресованого розміру”, доданого до заголовку кадру.

					ІАЛЦ. 045480.004 ПЗ	Арк.
						21
Змін.	Арк.	№ докум.	Підпис	Дата		

Прапорець “у” відповідає за надання інформації програмному забезпеченню про те, що кадр зашифровано. Якщо прапорець встановлено, до заголовку кадру додається один байт, який визначає, яким методом було зашифровано дані. Значення прапорця відповідають:

0 – кадр не зашифровано;

1 – кадр зашифровано.

Прапорець “у” відповідає за надання інформації програмному забезпеченню про те, чи належить цей кадр до групи з іншими кадрами. Якщо прапорець встановлено, до заголовку кадру додається ідентифікатор групи розміром в один байт. Значення прапорця відповідають:

0 – кадр не містить групової інформації;

1 – кадр містить інформацію про групу.

Усі невикористані біти прапорів повинні бути очищені.

Здебільшого, ці прапорці не використовуються в реальних файлах.

Розглянемо структуру кадру на прикладі реального файлу рис.2.2:

1. Червоним кольором виділено ідентифікатори заголовків, кожен з яких має довжину 4 байти. Перший заголовок має ідентифікатор «TALB», а другий – «TCON».

2. Зеленим кольором виділено розмір даних, записаних в кожен з кадрів. На розмір відведено по 4 байти для кожного з заголовків. Ці чотири байти представляють число. Для першого заголовку розмір буде 15(0000000f₁₆), а для другого – 13(0000000d₁₆).

3. Синім виділено по два поля прапорців для кожного з кадрів. В обох випадках всі прапорці є неактивними.

4. Фіолетовим виділено поля даних для кожного з кадрів. Для першого кадру поле даних займає 15 байтів, а для другого – 13. Саме ця довжина і записана в другому полі заголовку кадру.

00000200	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	49	44	33	03	00	00	00	00	1f	76	54	41	4c	42	00	00	ID3.....vTALB..
00000010	00	0f	00	00	01	ff	fe	10	04	3b	04	4c	04	31	04	3eяю...;.L.l.>
00000020	04	3c	04	54	43	4f	4e	00	00	00	0d	00	00	01	ff	fe	.<.TCON.....яю
00000030	21	04	42	04	38	04	3b	04	4c	04	54	49	54	32	00	00	!.B.8.;.L.TIT2..
00000040	00	0d	00	00	01	ff	fe	1d	04	30	04	37	04	32	04	30яю...0.7.2.0
00000050	04	54	50	45	31	00	00	00	17	00	00	01	ff	fe	12	04	.TPE1.....яю..
00000060	38	04	3a	04	3e	04	3d	04	30	04	32	04	35	04	46	04	8...>.=.0.2.5.F.
00000070	4c	04	54	50	4f	53	00	00	00	05	00	00	01	ff	fe	32	L.TPOS.....яю2
00000080	00	54	52	43	4b	00	00	00	0b	00	00	01	ff	fe	31	00	.TRCK.....яю1.
00000090	2f	00	31	00	34	00	54	59	45	52	00	00	00	0b	00	00	/.1.4.TYER.....
000000a0	01	ff	fe	32	00	30	00	32	00	30	00	55	53	4c	54	00	.яю2.0.2.0.USLT.
000000b0	00	00	20	00	00	01	65	6e	67	ff	fe	00	00	ff	fe	22engяю..яю"
000000c0	04	35	04	3a	04	41	04	42	04	20	00	3f	04	56	04	41	.5...A.B. .?.V.A
000000d0	04	3d	04	56	04	00	00	00	00	00	00	00	00	00	00	00	.=.V.....
000000e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Рисунок 2.2 – структура кадру метаданих формату ID3v2.3

2.3 Перелік типів кадрів, які найчастіше використовуються, їх ідентифікаторів, полів які вони зберігають, та їх призначення

2.3.1 Кадр унікального ідентифікатору файлу UFID

Призначення цього кадру – можливість ідентифікувати звуковий файл у базі даних, яка може містити більше інформації, що стосується вмісту. Структура кадру:

1. заголовок кадру з ідентифікатором є «UFID»;
2. ідентифікатор власника «текстовий рядок» \$00;
3. ідентифікатор (до 64 байт двійкових даних).

Заголовок містить ідентифікатор «UFID». Відразу за заголовком розміщується текстовий рядок, який має містити в собі адресу електронної пошти, або посилання на місце, де можна знайти адресу електронної пошти, що належить організації, відповідній за реалізацію конкретної бази даних. Це поле не може бути порожнім. Далі іде поле, що містить фактичний ідентифікатор розміром до 64 байтів. Тег може містити більше одного кадру UFID, але лише один з тим самим ідентифікатором власника.

2.3.2 Текстові кадри T000-TZZZ, за винятком TXXX

Текстові кадри є найважливішими, оскільки саме вони містять інформацію про виконавця, групу, альбом тощо. Тег може містити лише один інформаційний кадр для кожного з ідентифікаторів. Ідентифікатор текстового кадру починається з літери “Т”. Якщо за текстовим рядком йде розділювач, подальшу інформацію необхідно ігнорувати. Залежно від типу кодування розділювач може приймати наступні значення:

1. \$00 для кодування за стандартом ISO-8859-1;
2. \$00 00 для кодування за допомогою Unicode.

Структура кадру:

1. заголовок кадру з ідентифікатором у межах «T000»-«TZZZ», за винятком «TXXX»;
2. показник кодування тексту, який займає один байт;
3. рядок тексту, згідно з кодуванням.

В текстовому кадрі текстовий рядок зберігається згідно з одним з двох варіантів кодування. Цими варіантами є :

1. стандарт ISO-8859-1, який для зберігання кожного окремого символу використовує один байт;
2. Unicode, який для зберігання кожного окремого символу використовує два байти.

Поле показнику кодування тексту може приймати наступні значення:

1. \$00 – для кодування за стандартом ISO-8859-1;
2. \$01 – для кодування за допомогою Unicode.

Оскільки текстові кадри є найважливішими, розберемо структуру текстового кадру на конкретному прикладі рис.2.3:

1. сірим кольором виділено заголовок кадру, перші чотири символи якого мають значення «TPE1», що означає, що цей кадр містить інформацію про головних виконавців.

2. червоним кольором виділено поле покажчику кодування, значення даного поля - \$01, що означає, що текстовий рядок закодований за допомогою Unicode.

3. зеленим кольором виділено текстовий рядок, який містить інформацію, згідно з кодуванням. В даному випадку, за допомогою Unicode, закодовано рядок «Виконавець».

00000140	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f	
00000000	49	44	33	03	00	00	00	00	1f	76	54	41	4c	42	00	00	ID3.....vTALB..
00000010	00	0f	00	00	01	ff	fe	10	04	3b	04	4c	04	31	04	3eяю...;.L.l.>
00000020	04	3c	04	54	43	4f	4e	00	00	00	0d	00	00	01	ff	fe	.<.TCON.....яю
00000030	21	04	42	04	38	04	3b	04	4c	04	54	49	54	32	00	00	!.B.8.;.L.TIT2..
00000040	00	0d	00	00	01	ff	fe	1d	04	30	04	37	04	32	04	30яю..0.7.2.0
00000050	04	54	50	45	31	00	00	00	17	00	00	01	ff	fe	12	04	.TPE1.....яю..
00000060	38	04	3a	04	3e	04	3d	04	30	04	32	04	35	04	46	04	8...>.=.0.2.5.F.
00000070	4c	04	54	50	4f	53	00	00	00	05	00	00	01	ff	fe	32	L.TPOS.....яю2
00000080	00	54	52	43	4b	00	00	00	0b	00	00	01	ff	fe	31	00	.TRCK.....яю1.
00000090	2f	00	31	00	34	00	54	59	45	52	00	00	00	0b	00	00	/.1.4.TYER.....
000000a0	01	ff	fe	32	00	30	00	32	00	30	00	55	53	4c	54	00	.яю2.0.2.0.USLT.
000000b0	00	00	20	00	00	01	65	6e	67	ff	fe	00	00	ff	fe	22еняю...яю"
000000c0	04	35	04	3a	04	41	04	42	04	20	00	3f	04	56	04	41	.5...A.B. .?.V.A
000000d0	04	3d	04	56	04	00	00	00	00	00	00	00	00	00	00	00	.=.V.....
000000e0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000f0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Рисунок 2.3 – Структура текстового кадру

Далі наведено ідентифікатори текстових кадрів, визначені форматом ID3v2.3, та їх призначення:

1. «TALB» - призначений для зберігання назви альбому. Зазвичай представлений у вигляді текстового рядка.
2. «TBPM» - призначений для зберігання музичного темпу. Є цілим числом і представлений у вигляді числового рядка.
3. «TCOM» - призначений для зберігання імен композиторів, які розділено символом «/».
4. «TCON» - призначений для зберігання типу вмісту. Може містити сумісні з ID3v1 жанри. Для цього він повинен містити строку у вигляді «(X)», де X – номер жанру, передбачений форматом ID3v1. Можливо використання декількох жанрів в одному кадрі. Для цього в кадр необхідно записати строку у

вигляді «(X) (Y)», де X і Y – необхідні жанри. Також може містити строку з повною назвою жанру.

5. «TCOP» - зберігає повідомлення про авторські права. Повинен починатися з року і п'яти пробілів. Призначений для власника авторських прав на оригінальний звук, на не на аудіо-файл. Відсутність кадру не означає, що звук є загальнодоступним надбанням. Це означає лише те, що інформація недоступна або видалена.

6. «TDAT» - призначений для зберігання дати запису у форматі ДДММ. Інформаційне поле даного кадру завжди містить чотири символи у вигляді числового рядку.

7. «TDLY» - призначений для зберігання кількості мілісекунд тиші між піснями у списку відтворення. Час у даному полі представлений у вигляді числового рядку.

8. «TENC» - призначений для зберігання імені особи чи організації, яка кодувала аудіо-файл.

9. «TEXT» - призначений для зберігання тексту піні та імен письменників, які розділені символом «/».

10. «TFLT» - призначений для уточнення типу звуку.

11. «TIME» - призначений для зберігання довжини запису у форматі ГГХХ. Інформаційне поле даного кадру завжди містить чотири символи у вигляді числового рядку.

12. «TIT1» - призначений для зберігання опису групи звукової інформації.

13. «TIT2» - призначений для зберігання фактичної назви твору.

14. «TIT3» - призначений для зберігання уточнення субтитрів або опису. Використовується для інформації, безпосередньо пов'язаної з вмістом заголовку.

15. «TKEY» - призначений для зберігання музичного ключа, з якого починається звук. Представлений у вигляді текстового рядка з максимальною довжиною в три слова. Описи ключів представлено документації формату ID3v2.

16. «TLAN» - призначений для зберігання мови тексту, який співається в аудіо. Мова представлена трьома символами відповідно до ISO-639-2. Можливо зберігання декількох мов в одному кадрі.

17. «TLEN» - призначений для зберігання довжини аудіо-файлу в мілісекундах. Дане поле представлене у вигляді числового рядку.

18. «TMED» - призначений для зберігання типу носія, який є джерелом звуку. Може бути текстовим рядком, або одним з попередньо визначений форматом ID3v2 типів носіїв. Якщо тип є попередньо визначеним, його необхідно записати у вигляді «(XXX)», де XXX – тип, згідно з форматом ID3v2.

19. «TOAL» - призначений для зберігання назви оригінального запису, або джерела звуку.

20. «TOFN» - призначений для зберігання бажаного імені файлу, оскільки деякі носії не дозволяють бажаної довжини файлу. Ім'я файлу залежить від регістру і включає його суфікс.

21. «TOLY» - призначений для зберігання текстів та імен письменників оригінального запису, якщо, наприклад, музика у файлі є перекладом раніше випущеної пісні. Розділені символом «/».

22. «TOPE» - призначений для зберігання імен виконавців оригінального запису, якщо, наприклад, музика у файлі є перекладом раніше випущеної пісні. Розділені символом «/».

23. «TORY» - призначений для зберігання року, коли вийшов оригінальний запис, якщо, наприклад, музика у файлі є перекладом раніше випущеної пісні. Форматується так само, як і кадр «TYER».

24. «TOWN» - призначений для зберігання імені власника або власника ліцензії цього файлу або його вмісту.

25. «TRE1» - призначений для зберігання головних виконавців. Розділені символом «/».

26. «TRE2» - призначений для зберігання додаткової інформації про виконавців, наприклад, назви гурту або оркестру.

27. «TRE3» - призначений для зберігання імені диригенту.

					ІАЛЦ. 045480.004 ПЗ	Арк.
						27
Змін.	Арк.	№ докум.	Підпис	Дата		

28. «TPE4» - призначений для зберігання додаткової інформації про людей, які працювали над реміксом або інтерпретацією іншого існуючого твору.

29. «TPOS» - призначений для зберігання інформації про частину набору, з якого походить аудіо. Використовується, якщо джерело, описане в кадрі «TALB», розділене на кілька носіїв, таких як, компакт-диски. Значення може бути розширене символом «/».

30. «TPUB» - призначений для зберігання назви видавця.

31. «TRCK» - призначений для зберігання номеру порядку аудіо-файлу на його початковому записі. Поле може бути розширене символом «/» та числом, яке позначає загальну кількість доріжок або елементів оригінального запису.

32. «TRDA» - призначений для зберігання додаткової інформації про дату запису. Може використовуватись у доповненні до кадрів «TYER», «TDAT» та «TIME».

33. «TRSN» - призначений для зберігання назви інтернет-радіостанції, з якої передавалось записане аудіо.

34. «TRSO» - призначений для зберігання імені власника інтернет-радіостанції, з якої передавалось записане аудіо.

35. «TSIZ» - призначений для зберігання розміру аудіо-файлу в байтах виключаючи розмір тегу ID3v2. Представлений у вигляді числового рядку.

36. «TSRC» - призначений для зберігання міжнародного стандартного коду запису (ISRC). Представлений у вигляді дванадцяти символів.

37. «TSSE» - призначений для зберігання опису використаного апаратного забезпечення, яким було здійснено кодування файлу, та його налаштування.

38. «TYER» - призначений для зберігання року запису. Представлений у вигляді 4 символів.

2.3.3 Текстовий кадр TXXX

Текстовий кадр TXXX призначений для зберігання інформації, визначеної користувачем. Тег може містити декілька кадрів з ідентифікатором «TXXX», але опис має бути унікальний для кожного з них. Структура кадру TXXX:

1. заголовок кадру з ідентифікатором «TXXX»;
2. покажчик кодування тексту, який займає один байт;
3. опис, який є текстовим рядком, закодованим згідно з одним з варіантів кодування, і повинен закінчуватись розділювачами \$00 – для кодування згідно зі стандартом ISO-8859-1, та \$00 00 - для Unicode;
4. значення, яке також є текстовим рядком, закодованим згідно з одним з варіантів кодування.

В реальних файлах використовується досить рідко, оскільки, не містить важливої інформації для ідентифікації музичного твору.

2.3.4 Кадри URL-посилань W000-WZZZ, за винятком WXXX

За допомогою цього типу кадрів до тегу можуть додаватись динамічні дані, наприклад, веб-сторінки з інформацією про гастролі, ціни тощо. Тег може містити лише один інформаційний кадр для кожного з ідентифікаторів. Всі ідентифікатори кадру URL-посилань починаються з символу «W». Якщо за текстовим рядком йде розділювач, подальшу інформацію необхідно ігнорувати. Варіанти типів розділювачів описані в розділі 2.3.3. Структура кадру URL-посилань має наступний вигляд:

1. заголовок кадру з ідентифікатором «W000»-«WZZZ», за винятком «WXXX»;
2. поле що містить URL-адресу, що представлена текстовим рядком.

000000a0	1e 31 00 21 00 31 00 34 00 34 39 43 32 00 00 00	01.7.1.4.11EK...
000000b0	0b 00 00 01 ff fe 32 00 30 00 32 00 30 00 57 4fяю2.0.2.0.WO
000000c0	41 46 00 00 00 0a 00 00 67 6f 6f 67 6c 65 2e 63	AF.....google.c
000000d0	6f 6d 57 58 58 58 00 00 00 0c 00 00 00 00 67 6f	omWXXX.....go
000000e0	6f 67 6c 65 2e 63 6f 6d 43 4f 4d 4d 00 00 00 1a	ogle.comCOMM....

Рисунок 2.4 – Структура кадру URL-посилання

Досить розповсюджений тип кадру. Розглянемо його структуру на конкретному прикладі рис.2.4:

1. сірим кольором виділено заголовок кадру, перші чотири символи якого мають значення «WOAF», що означає, що цей кадр містить URL-посилання на веб-сторінку файлу;
2. червоним кольором виділено поле URL-посилання у вигляді текстового рядку.

Далі наведено ідентифікатори кадрів URL-посилань, визначені форматом ID3v2.3, та їх призначення:

1. «WCOM» - призначений для зберігання URL-посилання на певну веб-сторінку, яка містить комерційну інформацію. Наприклад, де можна купити пісню. Тег може містити більше одного кадру з ідентифікатором «WCOM», але лише за умови різного вмісту в даних кадрах.
2. «WCOP» - призначений для зберігання URL-посилання на веб-сторінку, яка містить умови використання та право власності на файл.
3. «WOAF» - призначений для зберігання URL-посилання на веб-сторінку, яка відповідає даному файлу.
4. «WOAR» - призначений для зберігання URL-посилання на офіційну веб-сторінку виконавців. Тег може містити більше одного кадру з ідентифікатором «WOAR», якщо аудіо містить більше одного виконавця, але лише за умови різного вмісту в даних кадрах.
5. «WOAS» - призначений для зберігання URL-посилання на веб-сторінку джерела аудіо-файлу, наприклад, фільму.
6. «WORS» - призначений для зберігання URL-посилання на веб-сторінку інтернет-радіостанції.
7. «WPAU» - призначений для зберігання URL-посилання на веб-сторінку, яка буде обробляти процес оплати цього файлу.
8. «WPUB» - призначений для зберігання URL-посилання на веб-сторінку видавця.

2.3.5 Кадр URL-посилань WXXX

Кадр URL-посилань WXXX призначений для зберігання URL-посилань, що стуються аудіо-файлу, визначений користувачем. Тег може містити декілька кадрів з ідентифікатором «WXXX», але опис має бути унікальний для кожного з них. Варто зауважити, що URL-посилання завжди кодується за стандартом ISO-8859-1, тоді як опис може кодуватись, як за стандартом ISO-8859-1, так і за допомогою Unicode. Структура кадру WXXX:

1. заголовок кадру з ідентифікатором «WXXX»;
2. покажчик кодування тексту, який займає один байт;
3. опис, який є текстовим рядком, закодованим згідно з одним з варіантів кодування, і повинен закінчуватись розділювачами \$00 – для кодування згідно зі стандартом ISO-8859-1, та \$00 00 - для Unicode;
4. URL-посилання, закодоване за стандартом ISO-8859-1, яке представлене у вигляді текстового рядку.

Досить розповсюджений тип кадру. Розглянемо структуру даного кадру на конкретному прикладі рис.2.5:

1. сірим кольором виділено заголовок, перші чотири символи якого мають значення «WXXX»;
2. червоним кольором виділено поле покажчику кодування, що містить значення \$00, яке означає, що поле опису закодовано за стандартом ISO-8859-1.
3. зеленим кольором виділено поле опису, згідно з кодуванням. В даному випадку це поле містить порожній рядок.
4. синім кольором виділено поле URL-посилання, яке завжди закодоване за стандартом ISO-8859-1.

000000d0	6f 6d 57 58 58 58 00 00 00 0c 00 00 00 00 67 6f	omWXXX.....go
000000e0	6f 67 6c 65 2e 63 6f 6d 43 4f 4d 4d 00 00 00 1a	ogle.comCOMM....
000000f0	00 00 01 65 6e 67 ff fe 00 00 ff fe 1a 04 3e 04	...engяю...яю...>.
00000100	3c 04 35 04 3d 04 42 04 30 04 40 04 55 53 4c 54	<.5.=.B.0.@.USLT

Рисунок 2.5 – Структура кадру WXXX

2.3.6 Кадр списку залучених людей IPLS

Кадр списку залучених людей призначений для зберігання імен тих людей, які були залучені до створення музичного твору, та те чим вони займалися. Представляє набір рядків, кожен з яких містить ім'я окремої людини, та те що саме вона робила. Структура кадру IPLS:

1. заголовок кадру з ідентифікатором «IPLS»;
2. покажчик кодування тексту, який займає один байт;
3. список текстових рядків, відповідно до кодування.

Кадри даного типу досить рідко зустрічаються в реальних файлах.

2.3.7 Кадр списку кодів часових подій ETCSO

Кадр списку кодів часових подій дозволяє синхронізувати ключові події в пісні або звуці. Варто зазначити, що всі події мають іти в хронологічній послідовності. Тег може містити лише один кадр типу ETCSO. Структура кадру ETCSO:

1. заголовок кадру з ідентифікатором «ETCSO»;
2. покажчик формату часової позначки розміром 1 байт;
3. список типу подій та часових позначок.

Формат часової позначки може приймати два значення:

1. \$01 – абсолютний час, розміром 32 біти, який для позначення використовує MPEG кадри;
2. \$02 – абсолютний час, розміром 32 біти, який для позначення використовує мілісекунди.

Список типу подій та часових позначок виглядає приблизно таким чином:

1. тип події розміром в один байт;
2. часова позначка.

Типи передбачених подій:

1. \$00 – відступ (не має значення);
2. \$01 – кінець початкової тиші;

3. \$02 – початок вступу;
4. \$03 – початок основної частини;
5. \$04 – початок завершення пісні;
6. \$05 – кінець завершення пісні;
7. \$06 – початок вірша;
8. \$07 – початок рефрену;
9. \$08 – початок антракту;
10. \$09 – початок теми;
11. \$0A – початок варіації;
12. \$0B – зміна ключа;
13. \$0C – зміна часу;
14. \$0D – небажаний короткий шум;
15. \$0E – тривалий шум;
16. \$0F – кінець тривалого шуму;
17. \$10 – кінець вступу;
18. \$11 – кінець основної частини;
19. \$12 – кінець віршу;
20. \$13 – кінець рефрену;
21. \$14 – кінець теми;
22. \$15 - \$DF – зарезервовано для майбутнього використання;
23. \$E0 - \$EF – невизначена синхронізація 0-F;
24. \$F0 - \$FC – зарезервовано для майбутнього використання;
25. \$FD – кінець аудіо (початок тиші);
26. \$FE – кінець файлу;
27. \$FF – ще один байт подій слідує за даним.

Кадри даного типу дуже рідко зустрічаються в реальних файлах.

Змін.	Арк.	№ докум.	Підпис	Дата

2.3.8 Кадр несинхронізованого тексту пісні USLT

Кадр несинхронізованого тексту пісні містить тексти або транскрипцію вокалу. Тег може містити більше одного кадру USLT, але опис має бути унікальний для кожного з них. Структура кадру USLT:

1. заголовок кадру з ідентифікатором «USLT»;
2. показчик типу кодування тексту, що займає 1 байт;
3. три байти що позначають мову;
4. опис вмісту, представлений у вигляді текстового рядку з роздільником, відповідно до типу кодування;
5. текст, представлений у вигляді текстового рядку відповідно до типу кодування.

У тексті допускаються символи нового рядку. Даний карт є дуже розповсюдженим. Розглянемо його структуру на прикладі конкретного файлу рис.2.6:

1. сірим кольором виділено заголовок кадру з ідентифікатором «USLT»;
2. червоним кольором виділено показчик типу кодування, який має значення \$01, що позначає кодування за допомогою Unicode;
3. помаранчевим кольором виділено поле що позначає мову, яке має значення «eng»;
4. зеленим кольором виділено поле опису вмісту, яке в даному випадку є порожнім рядком, закодованим за допомогою Unicode;
5. синім кольором виділено безпосередньо текст, закодований за допомогою Unicode.

00000100	3c 04	35 04	3d 04	42 04	30 04	40 04	55 53	4c 54	<.5.=.B.0.@.USLT
00000110	00 00	00 72	00 00	01 65	6e 67	ff fe	00 00	ff fe	...r...engяя...яя
00000120	22 04	35 04	3a 04	41 04	42 04	20 00	3f 04	56 04	".5.:.A.B. .?.V.
00000130	41 04	3d 04	56 04	20 00	40 04	4f 04	34 04	3e 04	A.=.V. .@.0.4.>.
00000140	3a 04	20 00	3d 04	3e 04	3c 04	35 04	40 04	20 00	:. .=>.<.5.@. .
00000150	31 00	0d 00	0a 00	22 04	35 04	3a 04	41 04	42 04	1.....".5.:.A.B.
00000160	20 00	3f 04	56 04	41 04	3d 04	56 04	20 00	40 04	.?.V.A.=.V. .@.
00000170	4f 04	34 04	3e 04	3a 04	20 00	3d 04	3e 04	3c 04	0.4.>.:. .=>.<.
00000180	35 04	40 04	20 00	32 00	00 00	00 00	00 00	00 00	5.@. .2.....

Рисунок 2.6 – Структура кадру USLT

2.3.9 Кадр синхронізованого тексту пісні SYLT

Кадр синхронізованого тексту пісні є ще одним способом включення текстів у аудіо-файл. На відміну від USLT, текст у даному кадрі синхронізовано зі звуком. Даний кадр також може бути використаний для синхронізованого опису подій, що відбуваються на сцені або в музиці. Тег може містити більше одного кадру синхронізованого тексту, але опис та мова мають бути унікальним для кожного з них. Структура кадру SYLT:

1. заголовок кадру з ідентифікатором «SYLT»;
2. покажчик типу кодування тексту, що займає 1 байт;
3. три байти що позначають мову;
4. покажчик формату часової позначки, що займає 1 байт;
5. тип вмісту;
6. опис вмісту, представлений у вигляді текстового рядку з роздільником, відповідно до типу кодування;
7. текст, структуру якого буде описано далі.

Поле “тип вмісту” може приймати такі значення:

1. \$00 – інший вміст;
2. \$01 – текст;
3. \$02 – транскрипція;
4. \$03 – назва руху або його частини;
5. \$04 – якась подія;
6. \$05 – акорд;
7. \$06 – дрібниці.

Формат часової позначки описаний в розділі 2.3.7.

Текст записаний після опису вмісту складається з блоків, призначених для синхронізації:

1. текстовий рядок, який закінчується розділювачем, відповідно до кодування;
2. ідентифікатор синхронізації \$00, або \$00 00, залежно від типу кодування;

3. часова позначка.

Приклад для порівняння синхронізованого і несинхронізованого текстів:

Несинхронізований:

«текст пісні рядок 1» \$0A «текст пісні рядок 2»

Синхронізований:

«текст» \$00 \$xx \$xx «пісні» \$00 \$xx \$xx «рядок 1» \$00 \$xx \$xx \$0A

«текст» \$00 \$xx \$xx «пісні» \$00 \$xx \$xx «рядок 2» \$00 \$xx \$xx

Де “\$xx \$xx” – байти часових позначок, а байт “\$0A” – перехід на новий рядок.

Даний тип кадрів в реальних файлах зустрічається досить рідко.

2.3.10 Кадр коментарів COMM

Кадр коментарів призначений для будь-якої текстової інформації. Тег може містити більше одного кадру синхронізованого тексту, але опис та мова мають бути унікальний для кожного з них. Структура кадру COMM:

1. заголовок кадру з ідентифікатором «COMM»;
2. покажчик типу кодування тексту, що займає 1 байт;
3. три байти що позначають мову;
4. опис вмісту, представлений у вигляді текстового рядку з роздільником, відповідно до типу кодування;
5. фактичний текст коментаря, представлений у вигляді текстового рядку відповідно до типу кодування;

Досить розповсюджений. В програмах, які не дають змоги редагувати кадр коментарів, в даний кадр, зазвичай записується назва програм, яка використовувалась, або порожні поля. Розглянемо його структуру на прикладі конкретного файлу рис.2.7:

1. сірим кольором виділено заголовок кадру з ідентифікатором «COMM»;

2. червоним кольором виділено показник типу кодування, який має значення \$01, що позначає кодування за допомогою Unicode;
3. помаранчевим кольором виділено поле що позначає мову, яке має значення «eng»;
4. зеленим кольором виділено поле опису вмісту, яке в даному випадку є порожнім рядком, закодованим за допомогою Unicode;
5. синім кольором виділено безпосередньо текст коментаря, закодований за допомогою Unicode.

00000000	6f 6d 5f 58 58 58 00 00 00 0c 00 00 00 00 6f 6f	оптхххх.....go
000000e0	6f 67 6c 65 2e 63 6f 6d 43 4f 4d 4d 00 00 00 1a	ogle.comCOMM....
000000f0	00 00 01 65 6e 67 ff fe 00 00 ff fe 1a 04 3e 04	...engяю...яю...>.
00000100	3c 04 35 04 3d 04 42 04 30 04 40 04 55 53 4c 54	<.5.=.B.0.@.USLT
00000110	00 00 00 72 00 00 01 65 6e 67 ff fe 00 00 ff fe	x enggg gg

Рисунок 2.7 – Структура кадру COMM

2.3.11 Кадр вкладеного зображення APIC

Кадр вкладеного зображення призначений для зображення, безпосередньо пов'язаного з аудіо-файлом. Формат зображення представлений у вигляді типу і підтипу MIME для зображення. У випадку, якщо ім'я типу пропущене, буде матися на увазі «image/». Опис – короткий опис малюнка, представлений у вигляді текстового рядку. Опис має максимальну довжину в 64 символи, але може бути порожнім. Тег може містити декілька кадрів APIC, але опис кожного має бути унікальним. Тег може містити лише один кадр з типом зображення \$01 або \$02. Замість даних зображення можна покласти URL-посилання на нього, задавши тип MIME рядком «-->». Використання посилань на картинки не рекомендується. Структура кадру APIC:

1. заголовок кадру з ідентифікатором «APIC»;
2. показник типу кодування тексту, що займає 1 байт;
3. MIME-тип, представлений у вигляді текстового рядку, за яким слідує розділювач \$00;

Змін.	Арк.	№ докум.	Підпис	Дата

4. тип зображення, представлений одним байтом;
5. опис зображення, представлений у вигляді текстового рядку з роздільником, відповідно до типу кодування;
6. зображення, представлене у вигляді набору двійкових даних.

Поле тип зображення може приймати наступні значення:

1. \$00 – інше зображення;
2. \$01 – значок файлу 32x32 пікселів (лише розширення PNG);
3. \$02 – інший значок файлу;
4. \$03 – передня обкладинка;
5. \$04 – задня обкладинка;
6. \$05 – листівка;
7. \$06 – медіа (наприклад сторона компакт-диску з зображенням);
8. \$07 – соліст, ведучий виконавець тощо;
9. \$08 – виконавець;
10. \$09 – диригент;
11. \$0A – гурт, оркестр тощо;
12. \$0B – композитор;
13. \$0C – автор тексту;
14. \$0D – місце запису;
15. \$0E – під час запису;
16. \$0F – під час виконання;
17. \$10 – знімок з фільму або відео;
18. \$11 – “A bright coloured fish”;
19. \$12 – ілюстрація;
20. \$13 – логотип групи або виконавця;
21. \$14 – логотип видавця або студії;

Кадри даного типу є дуже популярним. В більшості аудіо-файлів використовується лише одне зображення. Також більшість програм відображають лише одне зображення.

Розглянемо структуру кадру зображення на прикладі конкретного файлу рис.2.8:

1. сірим кольором виділено заголовок кадру з ідентифікатором «APIC»;
2. червоним кольором виділено показник типу кодування, який має значення \$00, що позначає кодування за стандартом ISO-8859-1;
3. помаранчевим кольором виділено поле типу MIME, представлене у вигляді текстового рядку, і має значення «image/jpg» з розділювачем \$00 в кінці рядку;
4. зеленим кольором виділено поле типу зображення, представлене у вигляді байту зі значенням \$03, який означає, що зображення – передня обкладинка;
5. синім кольором виділено опис зображення, який в даному випадку є порожнім рядком згідно з кодуванням ISO-8859-1;
6. фіолетовим кольором виділено частину двійкової інформації конкретного зображення у форматі jpg.

000000a0	fe 34 00 54 59 45 52 00 00 00 0b 00 00 01 ff fe	я4.TYER.....яю
000000b0	32 00 30 00 31 00 39 00 41 50 49 43 00 03 67 52	2.0.1.9.APIC..gR
000000c0	00 00 00 69 6d 61 67 65 2f 6a 70 67 00 03 00 ff	...image/jpg...я
000000d0	d8 ff e0 00 10 4a 46 49 46 00 01 01 01 00 60 00	Шаа..JFIF.....`.
000000e0	60 00 00 ff db 00 43 00 08 06 06 07 06 05 08 07	`...яН.С.....
000000f0	07 07 09 09 08 0a 0c 14 0d 0c 0b 0b 0c 19 12 13 \$.' "
00000100	0f 14 1d 1a 1f 1e 1d 1a 1c 1c 20 24 2e 27 20 22 \$.' "
00000110	2c 23 1c 1c 28 37 29 2c 30 31 34 34 34 1f 27 39	,#..(7),01444.'9
00000120	3d 38 32 3c 2e 33 34 32 ff db 00 43 01 09 09 09	=82<.342яН.С....

Рисунок 2.8 – структура кадру APIC

Висновки

В даному розділі було проаналізовано структуру тегу метаданих формату ID3v2.3. Було з'ясовано, що він складається з заголовку, який містить ідентифікатор тегу «ID3», прапорці, та загальний розмір тегу – 10 байтів заголовку. Тіло тегу складається з кадрів, кожен з яких має свій заголовок.

Кадри можуть бути різної довжини та містити різну інформацію. Ця інформація і є метаданими. Було проаналізовано типи кадрів, які використовуються в реальних файлах. Також було розглянуто на конкретних прикладах структури кадрів, які використовуються найчастіше.

					ІАЛЦ. 045480.004 ПЗ	Арк.
						40
Змін.	Арк.	№ докум.	Підпис	Дата		

3. ОПИС АЛГОРИТМІВ ЧИТАННЯ ТА ЗАПИСУ ТЕГІВ ID3v2.3, РЕАЛІЗАЦІЯ РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Алгоритм читання тегів

Читання тегу ID3v2.3 є однією з основних задач програмного забезпечення, розробленого в ході виконання дипломного проєкту. Для читання тегу і його полів необхідно мати структуру даних, в яку будуть записані дані з тегу. Спочатку необхідно визначити, прикріплено тег до файлу, чи ні. Це можна визначити, прочитавши з файлу перші три байти. Якщо вони містять дані «ID3», то файл містить тег формату ID3v2. Далі необхідно зчитати з файлу два байти. Інформацією, записаною в них є версія та ревізія тегу. В програмному забезпеченні, розробленому в ході виконання дипломного проєкту було прийняте рішення для аналізу метаданих, збережених за допомогою найпопулярнішої версії тегу формату ID3v2, а саме - третьої. Ревізія тегу несуттєво впливає на сумісність тегів. Отже, байтами зчитаними з файлу після ідентифікатора «ID3» мають приймати \$03, і \$00, або інше, дозволене форматом. За цими даними зчитується поле прапорців, яке має розмір один байт. Залежно від встановлених прапорців поводитьсь подальший аналіз вмісту тегу. Оскільки в реальних файлах прапорці майже не використовуються, програмне забезпечення було розроблене з підтримкою лише всіх скинутих прапорців. За байтом прапорців зчитується розмір тегу, закодований в чотирьох байтах, старші біти кожного з яких мають мати значення 0. Якщо байт має значення більше \$80, це означає помилку в структурі тегу, і його необхідно ігнорувати. Далі в циклі необхідно зчитувати заголовок кадру. Спочатку зчитуються чотири байти. Вони мають містити ідентифікатор кадру. Якщо зчитані байти не є ідентифікатором файлу, цикл має припинитися з помилкою. Якщо зчитані байти є ідентифікатором, далі зчитується розмір кадру, закодований в чотирьох байтах. Після поля з розміром кадру зчитується поле прапорців кадру, розміром два байти. Дані прапорці також використовуються дуже рідко, тому в розробленому програмному забезпеченні було прийнято рішення ігнорувати кадр у випадку встановлення прапорців кадру. Після зчитування заголовку кадру, в залежності

					ІАЛЦ. 045480.004 ПЗ	Арк.
						41
Змін.	Арк.	№ докум.	Підпис	Дата		

від типу кадру, та у випадку відсутності помилок мають зчитуватися поля кадру. В розробленому програмному забезпеченні, в залежності від значення першої літери ідентифікатору кадру, відбувається гілкування за допомогою оператора гілкування Switch. У випадку невідомого ідентифікатору програма переходить до гілки default, де вміст кадру пропускається. Дані полів, зчитаних з файлу, заносяться в поля структури даних ID3v_Tag. Схему алгоритму наведено в додатку А.

В розробленому програмному забезпеченні зчитування даних тегу є частиною процесу зчитування всіх даних файлу. Після виконання зчитування даних тегу зчитуються дані звуку.

3.2 Алгоритм запису тегів

Запис тегу ID3v2.3 є однією з основних задач програмного забезпечення, розробленого в ході виконання дипломного проєкту. Запис починається з запису ідентифікатора, який призначений для ідентифікації тегу ID3v2, а саме – трьох байтів, що містять значення «ID3». Далі записуються версія та ревізія тегу ID3v2. Розроблена програма використовує версію 3 та ревізію 0. Отже, записуються два байти зі значеннями \$03 і \$00. Далі записуються чотири байти зі значеннями \$00. Це розмір тегу, який рахується під час процедури запису даних у файл. Після запису останнього поля тегу, вказівник файлу переміщується на шостий байт і в чотири байти записується кодований розмір тегу. Кадри записуються в порядку, визначеному текстом програми. Спочатку записується ідентифікатор кадру у вигляді чотирьох байтів. Далі записуються чотири байти розміру, який займають інформаційні поля кадру. Розмір може бути різним, і залежить від типу кадру та розміру його полів. Далі записуються два байти прапорців, значення яких дорівнюють \$00. Після запису заголовку кадру записуються поля кадрів, згідно з форматом ID3v2.3. Першими записуються текстові кадри. Останнім записується кадр APIC. Схему алгоритму наведено в додатку Б.

В розробленому програмному забезпеченні запис даних є частиною процесу створення і запису нового аудіо-файлу.

3.3 Загальний опис розробленого програмного забезпечення

Програмне забезпечення, розроблене в ході виконання даного дипломного проєкту дозволяє користувачу швидко та зручно редагувати метадані формату ID3v2.3 аудіо-файлів, а також каталогізувати аудіо-файли. Програма має зрозумілий і швидкий у використанні користувацький інтерфейс, зображений на рисунку 3.1.

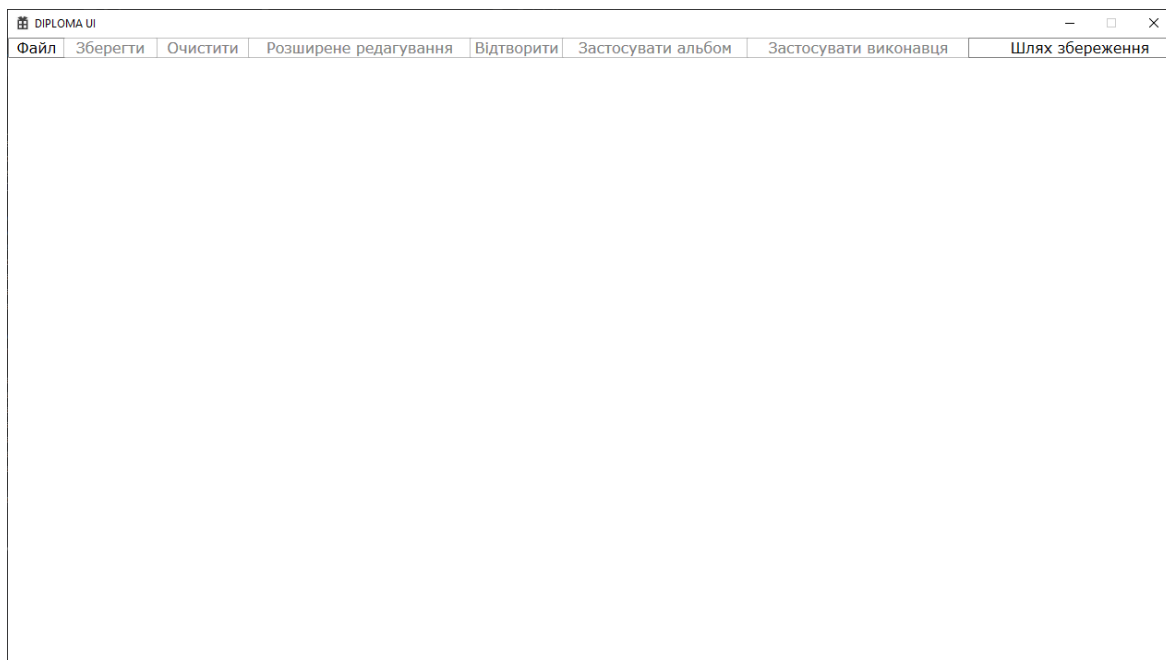


Рисунок 3.1 – Користувацький інтерфейс розробленої програми

Завантаження файлів відбувається за допомогою вбудованого програмне забезпечення файлового провідника, зображеного на рисунку 3.2. В файловому провіднику для завантаження можна виділяти кілька файлів. Було розроблено можливість виділення за допомогою клавіш “Shift” та “Ctrl”. Дуже зручні функції виділення всіх файлів від останнього до поточного, так виділення кількох файлів. Для завантаження файлів необхідно вибрати файли і натиснути кнопку “Відкрити”, як зображено на рисунку 3.3. Завантажені файли будуть відображені на головному вікні програми, зображеному на рисунку 3.4. З головного вікна

програми можна редагувати чотири основні поля даних аудіо-файлу, а саме: назва пісні, назва групи або ім'я виконавця, назва альбому та обкладинка, які зображено на рисунку 3.4.

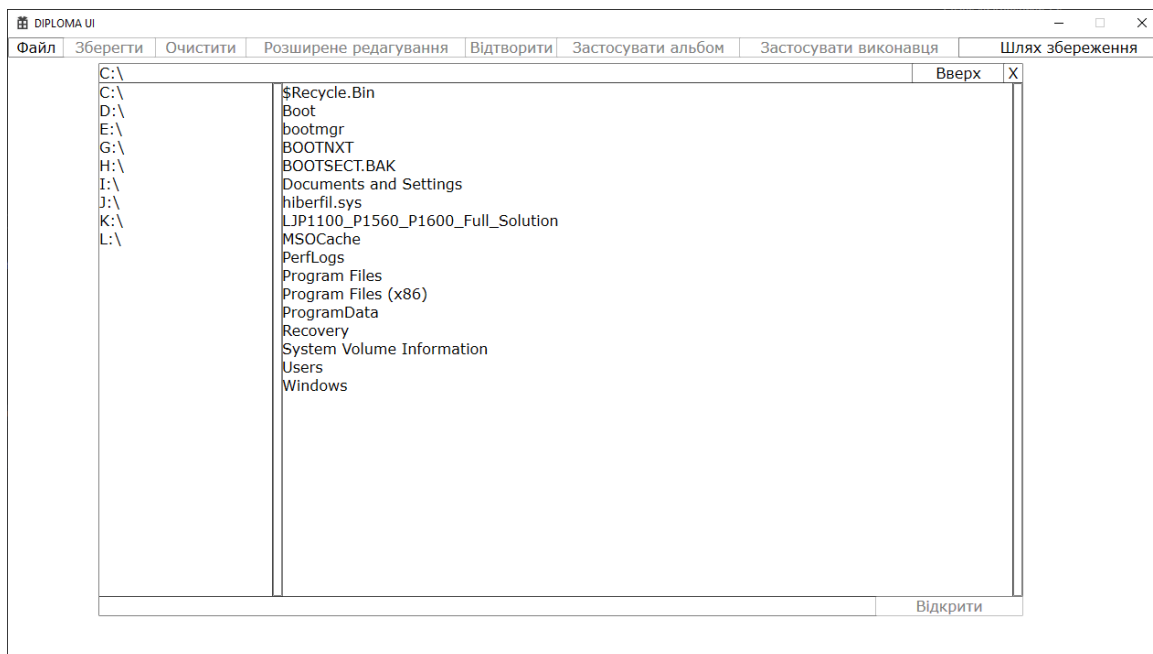


Рисунок 3.2 – Файловий провідник, вбудований в розроблене програмне забезпечення

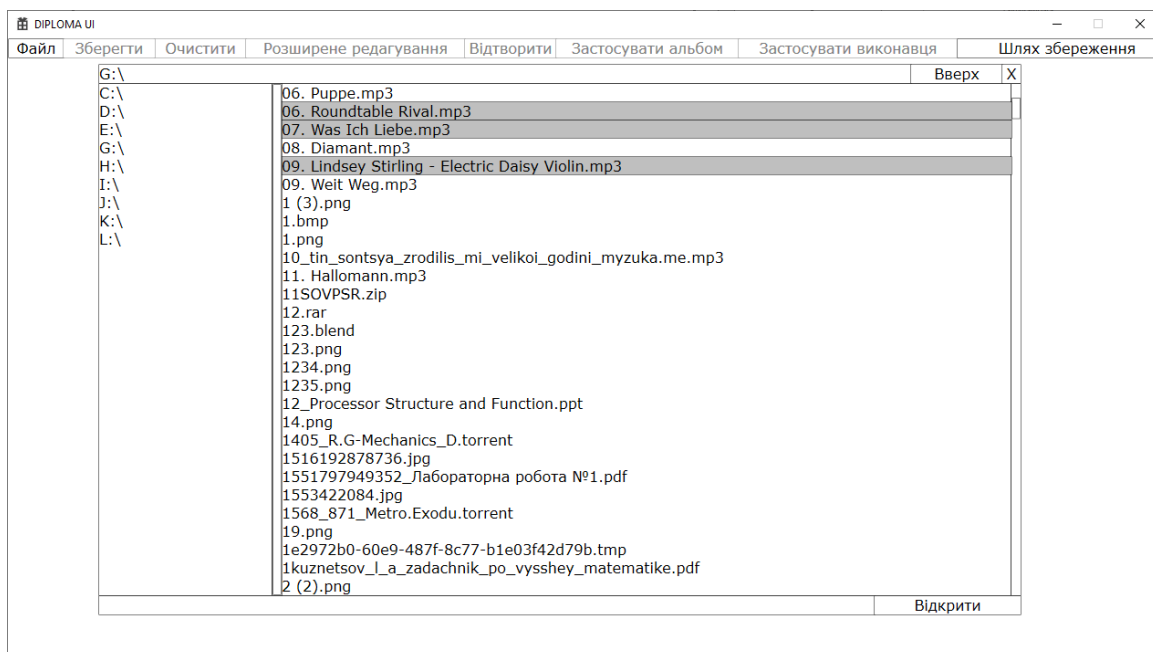


Рисунок 3.3 – Завантаження музичних файлів

Для завершення редагування текстових полів треба натиснути лівою кнопкою миші за полем, яке було редаговано, або натиснути клавішу "Enter".

Для редагування обкладинки необхідно натиснути на зображення, після чого вибрати нове зображення за допомогою файлового провідника та натиснути кнопку “Відкрити”. Для збереження редагованих файлів, необхідно вибрати, за яким способом будуть зберігатися файли за допомогою випадаючого меню “Шлях збереження”, яке зображено на рисунку 3.5, та натиснути кнопку “Зберегти”.

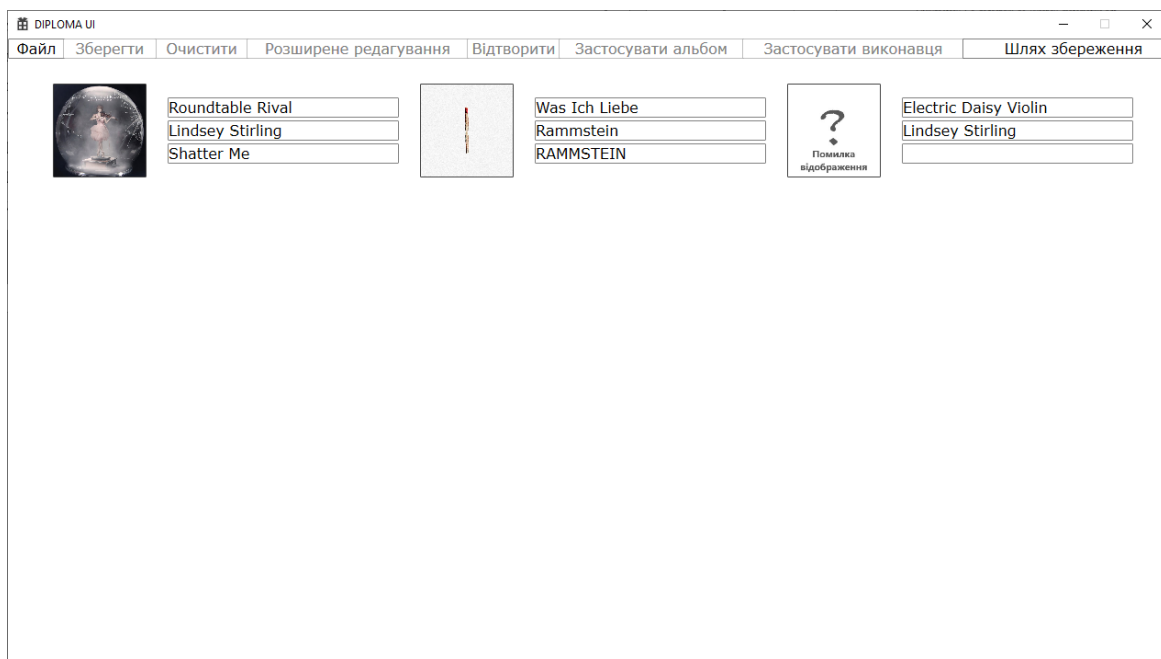


Рисунок 3.4 – Завантажені аудіо-файли

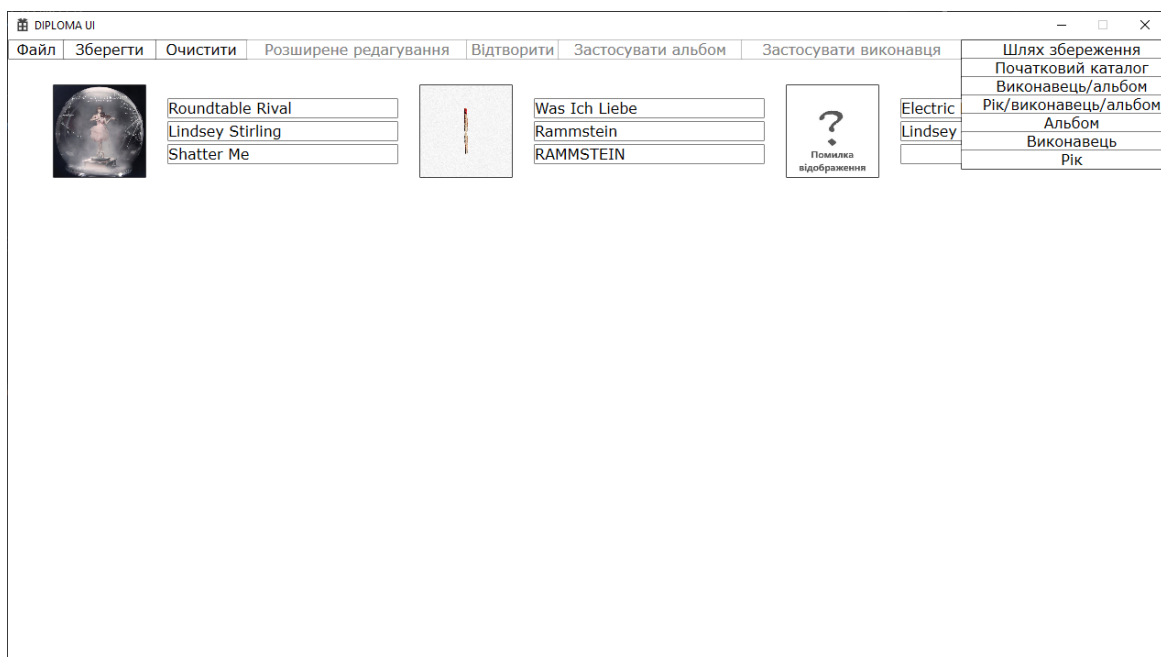


Рисунок 3.5 – Випадаюче меню “Шлях збереження”

Для того щоб мати змогу редагувати додаткові поля, такі як, рік запису аудіо-файлу, його номер, номер диску, жанр, деякі інші текстові метадані, коментар, та текст пісні, було розроблено інтерфейс для розширеного редагування метаданих. Щоб перейти до розширеного редагування необхідно вибрати один з аудіо-файлів в головному вікні програми та натиснути кнопку “Розширене редагування”, як зображено на рисунку 3.6. Після цього відкриється інтерфейс для розширеного редагування, що зображено на рисунку 3.7.

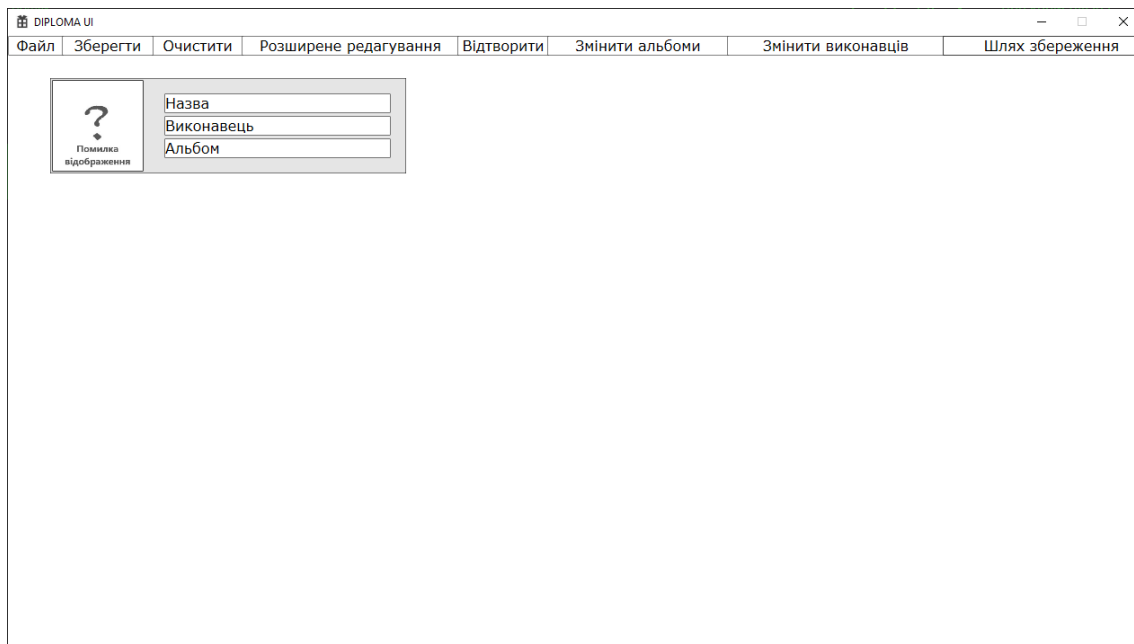


Рисунок 3.6 – Вибір аудіо-файлу

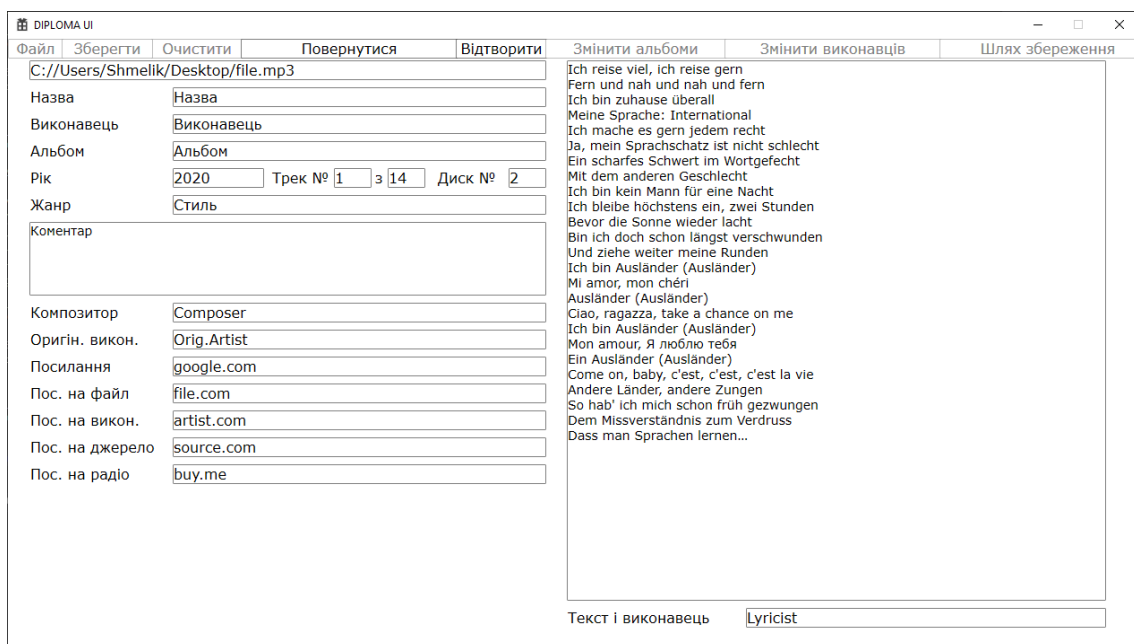


Рисунок 3.7 – Інтерфейс для розширеного редагування метаданих

Після закінчення редагування необхідно натиснути кнопку “Повернутися”. Програма повернеться до основного вікна інтерфейсу, і збереже аудіо-файл у тимчасову директорію. Для збереження у конкретну папку необхідно вибрати варіант каталогізації, натиснути кнопку зберегти і вибрати директорію для збереження.

3.4 Структури даних, використані в розробленому програмному забезпеченні

Модулі “ID3v_Tag” та “Track” містять класи, необхідні для зберігання і обробки інформації, пов’язаної з аудіо-файлами.

Основною оброблюваною інформацією в розробленому програмному забезпеченні є клас під назвою “Track”. Він містить такі поля:

- Name типу wstring, призначений для зберігання повного ім’я аудіо-файлу;
- Tag типу вказівника на екземпляр класу типу ID3v_Tag для зберігання метаданих;
- Folder_path типу filesystem::path для зберігання шляху до файлу;
- Data типу Vector<unsigned char> для зберігання даних аудіо-файлу;
- Data_size типу int для зберігання розміру даних аудіо-файлу.

Клас “ID3v_tag” містить досить велику кількість полів різних типів. Більшість з цих полів є полями для зберігання інформації кадрів метаданих.

Основними є текстові кадри та кадри для зберігання зображень. Змінна, яка містить текстові кадри має назву Text_Frames. Вона представлена у вигляді хеш-таблиці, ключем якої є назва текстового кадру, а даними – вказівник на екземпляр класу типу Text_Frame. Даний клас містить в собі поле Header, яке містить інформацію про заголовок та поле Data, яке містить в собі текстову інформацію. Для коректного збереження та відображення текстової інформації використано клас wstring, який дозволяє містити рядкову інформацію, що зберігається у вигляді символів розміром 2 байти. Змінна, яка містить кадри для

зберігання зображень називається APIC. Вона представлена у вигляді вектору вказівників на екземпляр класу типу Picture_Frame. Даний клас містить в собі такі поля: заголовок кадру під назвою Header, поле опису вмісту під назвою Description, поле яке містить MIME-тип під назвою MIME_Type, поле яке містить вказівник на двійкові дані картинки під назвою Data, та поле їх розміру під назвою Data_Size.

У якості графічної бібліотеки було використано Direct2D. За її допомогою було реалізовано клас під назвою Graphics, який містить в собі всі необхідні функції для відображення, а саме:

- відображати примітиви, такі як лінії, прямокутники тощо
- відображати текстову інформацію;
- відображати зображення у форматах jpeg та png.

Екземпляр даного класу створюється і ініціалізується під час початку роботи програми. Процес ініціалізації включає в себе такі процеси:

- створення об'єкту відображення, який відповідає за відображення у вікно програми;
- створення стилів відображення прямих;
- створення стилів відображення тексту;
- завантаження зображення, яке відображається у випадку, коли було зроблено спробу відобразити зображення, яке було завантажене з помилкою.

Модуль “Menue” містить у собі реалізацію класів, за допомогою яких було створено користувацький інтерфейс. Даний модуль містить класи, які реалізують кнопки, текстові поля, випадаючі меню, таблиці тощо.

Клас “file_system” дає змогу працювати з файловою системою. На його основі написаний вбудований в програму файловий провідник. Використовує класи, описані в просторі імен “filesystem”, який включено до стандарту C++17.

Клас “file_system_UI” містить реалізацію користувацького інтерфейсу, за допомогою якого, можна взаємодіяти з класом “file_system”.

Клас “main_window_UI” містить реалізацію користувацького інтерфейсу

					ІАЛЦ. 045480.004 ПЗ	Арк.
						48
Змін.	Арк.	№ докум.	Підпис	Дата		

для всієї програми.

Висновок

В даному розділі було розглянуто алгоритми читання та запису метаданих формату ID3v2.3, що входять до складу аудіо-файлів формату MP3. За розглянутими алгоритмами було написано програму для читання, редагування та запису метаданих обраного формату. Також було розроблено структури даних для зберігання метаданих. За допомогою бібліотеки для розробки 2D застосунків Direct2D було реалізовано користувацький інтерфейс. За допомогою бібліотеки filesystem було реалізовано можливість взаємодії з файловою системою комп'ютера.

Також було продемонстровано можливості користувацького інтерфейсу.

					ІАЛЦ. 045480.004 ПЗ	Арк.
						49
Змін.	Арк.	№ докум.	Підпис	Дата		

4. ТЕСТУВАННЯ І АНАЛІЗ РЕЗУЛЬТАТІВ

Для тестування розробленого програмного забезпечення було обрано три типи аудіо-файлів:

- аудіо-файли, які вже містять в собі метадані формату ID3v2.3;
- аудіо-файли, які містять не повну інформацію;
- аудіо-файли, без тегу метаданих.

При завантаженні у програму, аудіо-файли які не містять в собі метаданих формату ID3v2.3 не надають назви композиції, тому в це поле автоматично записується назва файлу, для можливості подальшої ідентифікації файлу, а поля, що містять ім'я виконавця та назву альбому залишаються порожніми. Якщо ж завантажений файл має вбудований тег метаданих – поля автоматично відображають інформацію, записану в аудіо-файл, за її наявності. Так, наприклад, тег метаданих може містити не всі поля. Приклад завантажених файлів показаний на рисунку 4.1:

1. перший аудіо-файл не містить тегу метаданих;
2. другий – містить тег метаданих, в якому записано назву композиції, та ім'я виконавця, але не записано назви альбому та зображення;
3. третій – містить тег метаданих, в якому записано три основні текстові поля і зображення.

Відредагуємо кожен з завантажених файлів наступним чином: завантажимо зображення для файлів, у яких воно відсутнє, встановимо текстові поля, які не мають даних, та змінимо назву композиції, аудіо-файл якої не містив тегу метаданих. Відредаговані дані зображено на рисунку 4.2.

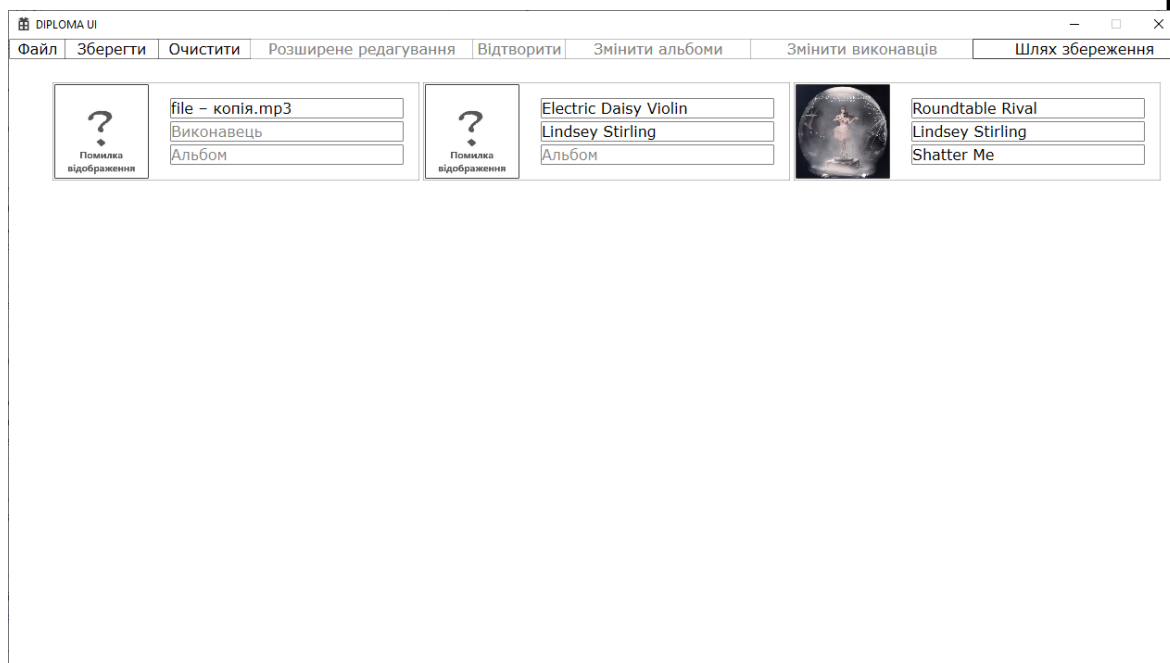


Рисунок 4.1 – Приклад з завантаженням аудіо-файлів з різними метаданими



Рисунок 4.2 – Відредаговані дані аудіо-файлів.

Для першого аудіо-файлу відредагуємо більшу кількість метаданих, за допомогою розширеного редагування, як показано на рисунку 4.3. Після закінчення розширеного редагування метаданих, перейдемо до головного вікна інтерфейсу, де оберемо варіант каталогізації “Рік/виконавець/альбом”, і збережемо аудіо-файли на диск G:/. Результатом роботи програми стануть нові

аудіо-файли з відредагованими тегами метаданих, збережені за обраним варіантом каталогізації, що показано на рисунку 4.4.

DIPLOMA UI							
Файл	Зберегти	Очистити	Повернутися	Відтворити	Змінити альбоми	Змінити виконавців	Шлях збереження
C://Users/Shmelik/Desktop/file – копія.mp3							
Назва	<input type="text" value="Ausländer"/>						
Виконавець	<input type="text" value="Rammstein"/>						
Альбом	<input type="text" value="Deutschland"/>						
Рік	<input type="text" value="2019"/>	Трек №	<input type="text" value="1"/>	з	<input type="text" value="5"/>	Диск №	<input type="text"/>
Жанр	<input type="text" value="Electro-industrial"/>						
<p>Відредаговано за допомогою розробленого програмного забезпечення.</p>							
Композитор	<input type="text"/>						
Оригін. викон.	<input type="text"/>						
Посилання	<input type="text"/>						
Пос. на файл	<input type="text"/>						
Пос. на викон.	<input type="text" value="https://www.rammstein.de/en/"/>						
Пос. на джерело	<input type="text"/>						
Пос. на радіо	<input type="text"/>						
				<p>Ich reise viel, ich reise gern Fern und nah und nah und fern Ich bin zuhause überall Meine Sprache: International Ich mache es gern jedem recht Ja, mein Schatz ist nicht schlecht Ein scharfes Schwert im Wortgefecht Mit dem anderen Geschlecht Ich bin kein Mann für eine Nacht Ich bleibe höchstens ein, zwei Stunden Bevor die Sonne wieder lacht Bin ich doch schon längst verschwunden Und ziehe weiter meine Runden Ich bin Ausländer (Ausländer) Mi amor, mon chéri Ausländer (Ausländer) Ciao, ragazza, take a chance on me Ich bin Ausländer (Ausländer) Mon amour, Я люблю тебя Ein Ausländer (Ausländer) Come on, baby, c'est, c'est la vie Andere Länder, andere Zungen So hab' ich mich schon früh gezwungen Dem Missverständnis zum Verdruss Dass man Sprachen lernen...</p>			
				<p>Текст і виконавець <input type="text" value="Till Lindemann"/></p>			

Рисунок 4.3 – Розширене редагування метаданих першого аудіо-файл

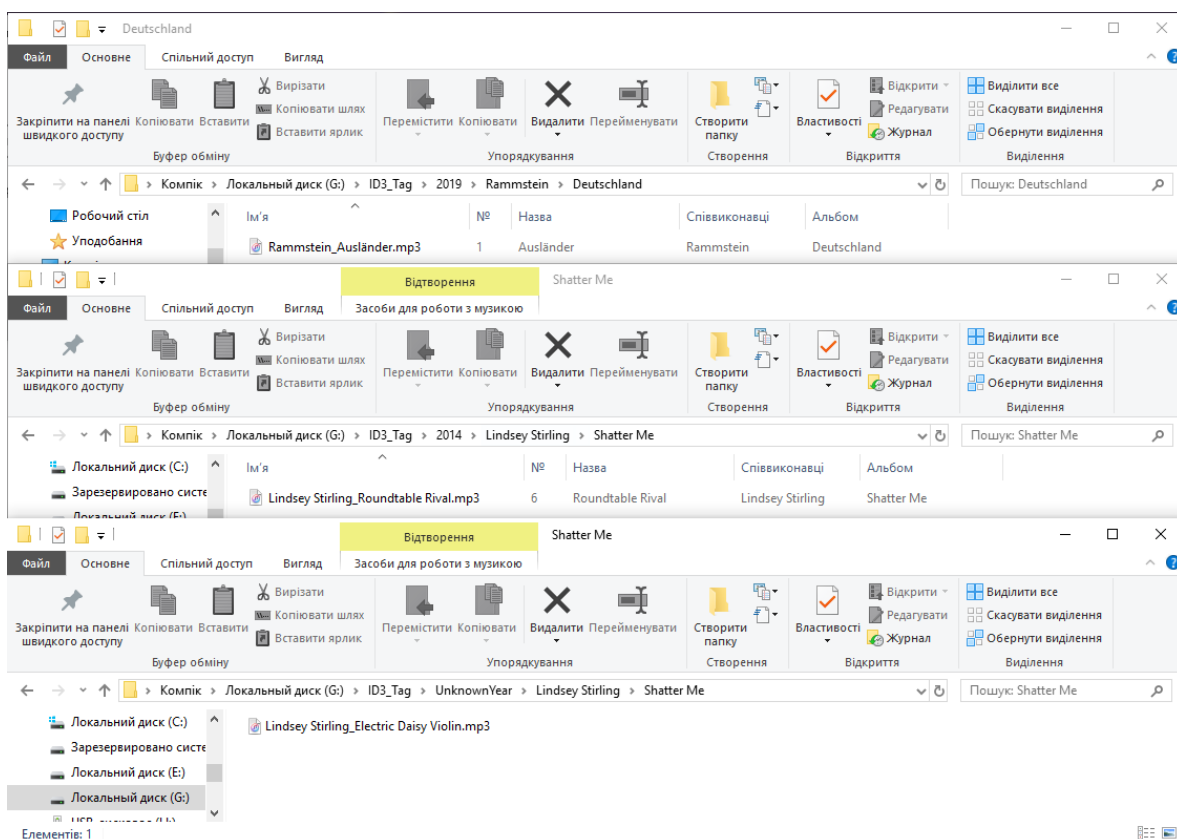


Рисунок 4.4 – Відредаговані аудіо-файли, збережені за обраним варіантом каталогізації

Висновок

В даному розділі було розглянуто приклад роботи з розробленим програмним забезпеченням на різних аудіо-файлах. Деякі аудіо-файли вже містять певну кількість метаданих, а деякі – не містять взагалі. Продемонстрована працездатність програмного забезпечення з обома варіантами аудіо-файлів. Також продемонстрована можливість розширеного редагування метаданих. Останнім продемонстровано можливість програмного забезпечення зберігати аудіо-файли згідно обраного варіанту каталогізації продемонстровано останнім.

					ІАЛЦ. 045480.004 ПЗ	Арк.
						53
Змін.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

Під час написання даного дипломного проєкту було проаналізовано існуючі програмні засоби для каталогізації аудіо-файлів та редагування їх метаданих. У кожного з цих засобів є свої переваги і недоліки. Було прийнято рішення про написання програмного засобу з відкритим кодом, зручним та зрозумілим інтерфейсом. Зважаючи на розповсюдженість форматів аудіо-файлів і метаданих, для розробки програмного забезпечення було вибрано формати MP3 та ID3v2.3. Також було проаналізовано структуру тегу формату ID3v2.3. За отриманими даними було розроблено алгоритми для зчитування, редагування і запису метаданих формату ID3v2.3. Використовуючи ці алгоритми, було розроблено програмне забезпечення для вирішення поставленої задачі. Розроблене програмне забезпечення було протестоване на аудіо-файлах формату MP3, як з наявним тегом метаданих, так і без нього. Результати тестів показали, що програмне забезпечення працює в обох випадках.

Розроблене в ході роботи над даним дипломним проєктом, програмне забезпечення може бути використане, як самостійний продукт, або для модифікації за своїм розсудом, завдяки відкритому коду. За допомогою відкритого коду можна впевнитись, що розроблене програмне забезпечення не містить прихованого шкідливого коду. Також, частина коду може бути використана для розробки або покращення іншого програмного забезпечення за даною тематикою.

Можливі модифікації:

- додавання підтримки інших форматів аудіо-файлів;
- додавання підтримки інших форматів метаданих;
- розширення можливостей користувацького інтерфейсу.

СПИСОК ВИКОРИСТНОЇ ЛІТЕРАТУРИ

1. Документація формату ID3v2.3: URL: <https://id3.org/id3v2.3.0> (дата звернення: 20.04.2020);
2. Офіційна документація бібліотеки Direct2D: URL: <https://docs.microsoft.com/en-us/windows/win32/direct2d/direct2d-portal> (дата звернення: 30.04.2020);
3. Офіційна документація бібліотеки filesystem: URL: <https://docs.microsoft.com/ru-ru/cpp/standard-library/filesystem?view=vs-2019> (дата звернення: 05.05.2020);
4. Цифрові звукові формати: URL: https://uk.wikipedia.org/wiki/Цифрові_звукові_формати (дата звернення: 27.04.2020);

Змін.	Арк.	№ докум.	Підпис	Дата

ІАЛЦ. 045480.004 ПЗ

Арк.

55